

Extraction d'Information Non Supervisée à Partir de Textes – Extraction et Regroupement de Relations entre Entités

Wei Wang

▶ **To cite this version:**

Wei Wang. Extraction d'Information Non Supervisée à Partir de Textes – Extraction et Regroupement de Relations entre Entités. Autre [cs.OH]. Université Paris Sud - Paris XI, 2013. Français. NNT : 2013PA112059 . tel-00998391

HAL Id: tel-00998391

https://theses.hal.science/tel-00998391

Submitted on 2 Jun 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.




UNIVERSITÉ DE PARIS-SUD

ÉCOLE DOCTORALE D'INFORMATIQUE DE PARIS-SUD

T H È S E

POUR OBTENIR LE TITRE DE

DOCTEUR EN SCIENCE
DE L'UNIVERSITÉ DE PARIS-SUD

MENTION : INFORMATIQUE

SOUTENUE LE 16 MAI 2013 PAR

WEI WANG

UNSUPERVISED INFORMATION EXTRACTION FROM TEXT - EXTRACTION AND CLUSTERING OF RELATIONS BETWEEN ENTITIES

RAPPORTEURS :	GAËL DIAS	-	GREYC
	PASCALE SÉBILLOT	-	IRISA/INSA DE RENNES
EXAMINATEURS :	MATHIEU ROCHE	-	LIRMM
	MICHÈLE SEBAG	-	LRI
DIRECTEUR :	BRIGITTE GRAU	-	LIMSI/ENSIIE
ENCADRANTS :	ROMARIC BESANÇON	-	CEA LIST
	OLIVIER FERRET	-	CEA LIST

© COPYRIGHT BY WEI WANG, 2013.
ALL RIGHTS RESERVED.

Acknowledgements

I am very grateful for the generous support from my three advisers Romaric Besançon, Olivier Ferret and Brigitte Grau in the past years. Without their attentive guide, patient listening and constructive criticism, this thesis can not be achieved. I would like to thank them sincerely.

I would like also to thank the members of the jury, Gaël Dias, Mathieu Roche, Michèle Sebag and Pascale Sébillot, for their careful reading and their instructive questions and suggestions.

Many thanks to colleges from the LVIC group: Gaël de Chalendar, Bertrand Delezoïde, Faïza Gara, Parick Hède, Benjamin Labbé, Hervé Le Borgne, Olivier Mesnard, Adrian Popescu and Nasredine Semmar. Thanks a lot for the help and kindness. I want also to thank all ancient members of the group during my presence, especially Ludovic Jean-Louis and Tonio Wandmacher. I am also thankful for the company of all the other PhD students and interns, in particular those who share the same office: Nicolas Ballas, Dhouha Bouamor and Amel Znaidia.

Last but not least, I would like to thank my family from the deep of my heart. Thank you very much for your endless support during all these years.

Abstract

Unsupervised information extraction in open domain gains more and more importance recently by loosening the constraints on the strict definition of the extracted information and allowing to design more open information extraction systems. In this new domain of unsupervised information extraction, this thesis focuses on the tasks of extraction and clustering of relations between entities at a large scale. The objective of relation extraction is to discover unknown relations from texts. A relation prototype is first defined, with which candidates of relation instances are initially extracted with a minimal criterion. To guarantee the validity of the extracted relation instances, a two-step filtering procedure is applied: the first step with filtering heuristics to remove efficiently large amount of false relations and the second step with statistical models to refine the relation candidate selection. The objective of relation clustering is to organize extracted relation instances into clusters so that their relation types can be characterized by the formed clusters and a synthetic view can be offered to end-users. A multi-level clustering procedure is designed, which allows to take into account the massive data and diverse linguistic phenomena at the same time. First, the basic clustering groups similar relation instances by their linguistic expressions using only simple similarity measures on a bag-of-word representation for relation instances to form high-homogeneous basic clusters. Second, the semantic clustering aims at grouping basic clusters whose relation instances share the same semantic meaning, dealing with more particularly phenomena such as synonymy or more complex paraphrase. Different similarities measures, either based on resources such as WordNet or distributional thesaurus, at the level of words, relation instances and basic clusters are analyzed. Moreover, a topic-based relation clustering is proposed to consider thematic information in relation clustering so that more precise semantic clusters can be formed. Finally, the thesis also tackles the problem of clustering evaluation in the context of unsupervised information extraction, using both internal and external measures. For the evaluations with external measures, an interactive and efficient way of building reference of relation clusters proposed. The application of this method on a newspaper corpus results in a large reference, based on which different clustering methods are evaluated.

Keywords: natural language processing, unsupervised information extraction, relation clustering, semantic similarity

Résumé

L'extraction d'information non supervisée en domaine ouvert est une évolution récente de l'extraction d'information adaptée à des contextes dans lesquels le besoin informationnel est faiblement spécifié. Dans ce cadre, la thèse se concentre plus particulièrement sur l'extraction et le regroupement de relations entre entités en se donnant la possibilité de traiter des volumes importants de données. L'extraction de relations se fixe ici pour objectif de faire émerger des relations de type non prédéfini à partir de textes. Elle est réalisée en deux temps : des relations candidates sont d'abord extraites sur la base de critères simples mais efficaces pour être ensuite filtrées selon des critères plus avancés. Ce filtrage associe lui-même deux étapes : une première étape utilise des heuristiques pour éliminer rapidement les fausses relations en conservant un bon rappel tandis qu'une seconde étape se fonde sur des modèles statistiques pour raffiner la sélection des relations candidates. Le regroupement de relations a quant à lui pour objectif d'organiser les relations extraites pour en caractériser le type et en offrir une vue synthétique. Il est réalisé dans le cas présent selon une stratégie multiniveau permettant de prendre en compte à la fois un volume important de relations et des critères de regroupement élaborés. Un premier niveau de regroupement, dit de base, réunit des relations proches de par leur expression linguistique grâce à une mesure de similarité vectorielle appliquée à une représentation de type ≪sac-de-mots≫ pour former des clusters fortement homogènes. Un second niveau de regroupement est ensuite appliqué pour traiter des phénomènes plus sémantiques tels que la synonymie et la paraphrase et fusionner des clusters de base recouvrant des relations équivalentes sur le plan sémantique. Ce second niveau s'appuie sur la définition de mesures de similarité au niveau des mots, des relations et des clusters de relations en exploitant soit des ressources de type WordNet, soit des thésaurus distributionnels. Un clustering des relations de nature thématique permet enfin d'améliorer la précision des clusters de relations formés en s'appuyant sur un contexte plus global. La thèse aborde également le problème de l'évaluation de l'extraction d'information non supervisée par l'entremise de mesures internes et externes. Pour les mesures externes, une méthode interactive est proposée pour construire manuellement un large ensemble de clusters de référence. Son application sur un corpus journalistique de grande taille a donné lieu à la construction d'une référence vis-à-vis de laquelle les différentes méthodes de regroupement proposées dans la thèse ont été évaluées.

Mots-clés : traitement automatique du langage naturel, extraction d'information non supervisée, regroupement de relations, similarité sémantique

Contents

Acknowledgements . . . iii
Abstract . . . v
Résumé . . . vii
List of Tables . . . xiii
List of Figures . . . xvii

Synthèse **xxi**
1 Introduction . . . xxi
2 Vue d'ensemble . . . xxii
3 Extraction de Relations Non Supervisée . . . xxv
4 Regroupement de Relations . . . xxxii
5 Résultats et Évaluations . . . xli
6 Travaux liés au clustering de relations . . . xlvii
7 Conclusion et perspectives . . . xlviii

1 Introduction **1**

2 State of The Art **7**
2.1 Overview of Information Extraction Evaluations . . . 8
2.1.1 MUC Series . . . 8
2.1.2 ACE Series . . . 9
2.1.3 TAC KBP Series . . . 9
2.1.4 TREC Series . . . 10
2.1.5 Evaluations in Specific Domains . . . 11
2.1.6 A Brief Summary . . . 11
2.2 Supervised Information Extraction . . . 13
2.3 Semi-supervised Information Extraction . . . 14
2.3.1 Bootstrapping from Selected Seeds: Snowball Effect . . . 15

2.3.2 Learning from Large Open Resources: Distant Learning 17
2.4 Unsupervised Information Extraction 19
2.4.1 Binary Relation Discovery with Clustering Algorithms 21
2.4.2 Query-oriented Unsupervised Information Extraction 25
2.4.3 Open Relation Extractors 27
2.4.4 Generative Models and Rule-based System 30
2.4.5 Complex Relation Extraction 32
2.4.6 Relation Organization 34
2.5 Overview of Our Unsupervised IE System 35
2.5.1 Positioning of The Thesis 35
2.5.2 Overview of The System 36

3 Relation Extraction **39**
3.1 Relation Definition 40
3.2 Terminology of Relation Extraction 41
3.3 Relation Characterization and Extraction 42
3.3.1 Relation Prototype 42
3.3.2 Initial Extraction of Relation Candidates 44
3.3.3 Error Analysis of Relation Instances 45
3.4 Filtering by Heuristics 47
3.5 Filtering by Machine Learning Models 49
3.5.1 Relation Annotation 50
3.5.2 Binary Classification Models for Relation Candidates 52
3.5.3 Sequential Model for Machine Learning Filtering 55
3.6 Comparison with Other Systems 59
3.7 Application of Relation Filtering 62
3.8 Conclusions and Perspectives 64

4 Relation Clustering **67**
4.1 Relation Clustering Problematic 68
4.1.1 Difficulties of Relation Clustering 68
4.1.2 Relation Clustering System Design 69
4.2 Similarity Measures for Clustering 73
4.2.1 Basic Similarity Measures 73
4.2.2 Semantic Similarities between Words 75
4.2.3 Similarity Matrix Calculation 77

4.3 Clustering Algorithms . . . 78
4.3.1 Markov Clustering . . . 80
4.3.2 Shared Nearest Neighbour Clustering . . . 82
4.3.3 Clustering Algorithm Choice . . . 83
4.4 Basic Clustering . . . 85
4.4.1 Term Weighting Strategies . . . 86
4.4.2 Labeling and Refinement of Basic Clusters . . . 88
4.5 Semantic Clustering . . . 90
4.5.1 Similarity Measures for Semantic Clustering . . . 90
4.5.2 Part-of-Speech Issues and Similarity Choice . . . 93
4.6 Topic-based Relation Clustering . . . 94
4.6.1 Topic Segmentation and Context Clustering . . . 95
4.6.2 Combination of Relation Clustering and Context Clustering . . . 96
4.7 A Summary of Our Clustering Approaches . . . 98

5 Evaluations and Results **99**
5.1 Clustering Evaluation Problems . . . 100
5.2 Clustering Evaluation Framework . . . 102
5.2.1 Internal Evaluation Measures . . . 102
5.2.2 External Evaluation Measures . . . 103
5.2.3 Reference Clusters Building . . . 107
5.2.4 The Outline of Experiments . . . 112
5.3 The Impact of Filtering Procedure on Relation Clustering . . . 114
5.3.1 Evaluation with Internal Measures . . . 114
5.3.2 Evaluation with External Measures . . . 115
5.4 Experiments of Basic Clustering . . . 119
5.4.1 Basic Clustering with Binary Weighting Configuration . . . 119
5.4.2 Comparison of Different Weighting Strategies . . . 121
5.4.3 Basic Clustering Results . . . 123
5.5 Experiments of Semantic Clustering . . . 125
5.5.1 Preliminary Experiments of Semantic Clustering . . . 126
5.5.2 Semantic Clustering Using The Best Basic Clusters . . . 131
5.5.3 Semantic Clustering Results . . . 135
5.5.4 The Effects of Multi-Level Clustering . . . 136
5.6 Experiments of Topic-based Relation Clustering . . . 138
5.6.1 Context Clustering . . . 138

5.6.2 Sequential Application of Context Clustering and Relation Clustering 139
5.6.3 Integration of Context Clusters and Relation Clusters 142
5.7 Conclusions and Perspectives 144

6 Conclusions and Perspectives **149**
6.1 A Brief Conclusion on Contributions 149
6.2 What Can Be Done Next? 152

A Overall Performance Estimation of Two-step Filtering **155**
A.1 Contingency Table 155
A.2 Approximation of Overall F-measures 157
A.3 Estimation of The Ratio of False Negative Decisions 160

Publications **161**

Bibliography **163**

List of Tables

1 Volumétrie des instances de relations par extraction initiale xxv
2 Effet de l'application des heuristiques de filtrage xxvi
3 Évaluation du filtrage par les heuristiques xxvii
4 Résultat de l'annotation manuelle des relations xxviii
5 Évaluation des classifieurs statistiques xxix
6 Évaluation du modèle CRF xxxi
7 Niveau de filtrage des relations à l'issue de chacune des étapes xxxii
8 Pondération grammaticale: distribution des poids selon la catégorie morpho-syntaxique xxxv
9 Impact du filtrage sur les résultats du regroupement des relations xlii
10 Résultats du clustering de base pour plusieurs pondérations en utilisant le Markov Clustering (MCL) et un premier regroupement par mots-clés xlii
11 Résultats du clustering sémantique xliii
12 Exemples de mots regroupés dans les clusters sémantiques xliv
13 Résultats du clustering thématique de relations xlv
14 Mots caractéristiques des clusters thématiques au niveau de la référence pour le type de relation *lead_by* xlvi

2.1 Evolution of information extraction evaluations 12
2.2 A variety of approaches for semi-supervised IE 19
2.3 Comparison of different options of binary relation clustering 24
2.4 The positioning of this thesis 36

3.1 Volume of extracted candidates of relation instances 45
3.2 Effects of the application of filtering heuristics on a sample of 8,000 relation candidates for each relation category 48
3.3 Evaluation of filtering heuristics for each relation category 49
3.4 Manual annotation of 200 relation candidates for each category 52

3.5 Corpus finally used for classifier training 53
3.6 Evaluation of statistical classifiers 55
3.7 Evaluation of statistical classifiers 57
3.8 Comparison for statistical classifiers using named entity types as features or not 59
3.9 Annotation of all extracted relation instances on the set of 500 sample sentences 60
3.10 Comparison between REVERB and our system: evaluation by the reference from REVERB systems 60
3.11 Comparison between REVERB and our system: evaluation by our annotated reference 62
3.12 Relation volumes after each filtering step 63
3.13 Overall F-measures estimation for two steps filtering 63

4.1 Time and space complexities for different clustering algorithms 84
4.2 Different categories of weighting by Part-of-Speech 89

5.1 Impact of the filtering procedure on clustering results: evaluation with internal measures *Expected density* and *Connectivity* 115
5.2 Impact of the filtering procedure on clustering results: global coverage of relation instances in reference that are in result clusters 115
5.3 Impact of the filtering procedure on clustering results: evaluation with external measures *Rand Index* and F-measures 116
5.4 Impact of the filtering procedure on clustering results: difference of the number of TN, FP, FN and TN decisions 117
5.5 Impact of the filtering procedure on clustering results: evaluation with external measures *Purity*, *Inverse Purity* and NMI 118
5.6 Cluster statistics of basic clustering with MCL and its refinement 124
5.7 Results of basic clustering applied on all relation instances 124
5.8 Characteristics of the clusters formed by semantic clustering using different similarities measures 130
5.9 The contribution of nouns in terms of cluster characteristics 134
5.10 Examples of semantic clustering results 135
5.11 Some examples of context clusters with their characteristic words, obtained by the MCL algorithm and a tf-idf weighting on segments' words 140

5.12 Results of applying relation clustering and context clustering sequentially in different orders . . . 141
5.13 Characteristic words of context clusters inside the thematic-specific reference for the relation *lead_by* . . . 145

A.1 Contingency table for classifier evaluation . . . 155
A.2 Two-level contingency table . . . 156
A.3 Estimation of T and T' . . . 159
A.4 Overall F-measures estimation for two-step filtering . . . 159

List of Figures

1 Exemple de relation extraite xxiii
2 Étiquetage des relations par un modèle CRF xxx
3 Regroupement des relations en niveaux xxxiii
4 Regroupement thématique des relations xl
5 Distribution des similarités entre les relations et entre les clusters de base . xlvii

1.1 Overview of the thesis 3

2.1 Template for an attack event from MUC-4 8
2.2 Relation types defined in ACE (Doddington et al., 2004) 9
2.3 Attributes for named entities in KBP 10
2.4 One topic example from TREC 2009, Entity track 11
2.5 Three categories of Information Extraction tasks 12
2.6 LDA model (Blei et al., 2003) 30
2.7 Overview of the system 36

3.1 Example of extracted relation 43
3.2 Two-step filtering for initially extracted relation candidates 47
3.3 Interface for the annotation of relation instances 50
3.4 Sequential representation of sentence annotation for a true relation 56
3.5 Sequential representation of sentence annotation for a false relation 56
3.6 Negative relation instances in REVERB reference 61
3.7 Examples of relation instances for the category PERSON-ORGANIZATION . 65

4.1 The procedure of multi-level relation clustering: basic clustering and semantic clustering 71
4.2 The procedure of context clustering 71
4.3 Overview of relation clustering and context clustering 72
4.4 A taxonomy of clustering algorithms 79

4.5 Markov Clustering procedure: evolution of edges (Van Dongen, 2000) . . . 81
4.6 Shared neighbours between two objects 82
4.7 Examples of variations of the linguistic expression of the $Cmid$ part of relation instances based on the verb "**retire**" 86
4.8 Topic-based relation clustering: the application of one clustering after another 97
4.9 Topic-based relation clustering: the integration of two kinds of clusters . . . 98

5.1 Reference assignment strategy for *purity* measure 105
5.2 Reference assignment strategy for *inverse purity* measure 105
5.3 An example of bootstrapping for building reference clusters 108
5.4 Interface of relation query tool 110
5.5 Reference cluster example for relation *grow_up_in* of the category PER–LOC 111
5.6 Experiments of basic clustering 112
5.7 Experiments of semantic clustering 113
5.8 Experiments of topic-based relation clustering 113
5.9 Performance comparison of basic clustering using different pruning thresholds with binary weighting MCL algorithm 120
5.10 Performance comparison of basic clustering using different inflation values with binary weighting MCL algorithm 121
5.11 Performance comparison of MCL algorithm using different weighting strategies for similarity calculation 122
5.12 Refinement of MCL results 123
5.13 Reference cluster example for relation *create* of the category ORG-ORG . . 125
5.14 Performance of *base-line* and *best-line* compared to results using binary weighting MCL 127
5.15 Performance different semantic similarities for preliminary experiments of semantic clustering 129
5.16 Performance of different semantic similarities for semantic clustering, detailed for different relation categories 130
5.17 Performance of semantic clustering using different semantic similarities, based on the best basic clusters 132
5.18 Evaluation of the contribution of nouns for semantic clustering 134
5.19 Distribution of similarities between relation instances (D) and between basic clusters (D') 137
5.20 Intersection of context clustering results with relation clusters 142

5.21 Intersection of context clustering results with relation clusters, for the relation “lead_by” in different topics 143

Synthèse

1 Introduction

Le domaine de l'Extraction d'Information (EI) s'est longtemps inscrit dans le paradigme établi par les conférences d'évaluation MUC (*Message Understanding Conference*) et poursuivi par des campagnes telles que ACE (*Automatic Content Extraction*). Les tâches définies par ces campagnes concernent l'extraction d'information supervisée, pour laquelle le type d'information à extraire est prédéfini et des instances sont annotées dans des corpus représentatifs. À partir de ces données, des systèmes conçus manuellement ou par apprentissage automatique peuvent être développés. Des approches semi-supervisées ont été définies plus récemment pour s'affranchir partiellement des contraintes de disponibilité de telles données. Par exemple, dans le cadre de la tâche KBP (*Knowledge Base Population*) de la campagne TAC (*Text Analysis Conference*), l'extraction de relations s'appuie sur une base de connaissances existante (construite à partir des infoboxes de Wikipédia), mais sans données annotées. Dans ce cas, des techniques de supervision distante Mintz et al. (2009) peuvent être appliquées. Ces méthodes semi-supervisées incluent également des techniques d'amorçage (*bootstrapping*) Grishman and Min (2010) permettant de s'appuyer sur un nombre limité d'exemples pour en extraire d'autres, comme par exemple dans Brin (1998) pour extraire une relation entre un livre et son auteur.

L'extraction d'information non supervisée diffère de ces tâches en ouvrant la problématique de l'extraction de relations à des relations de type inconnu *a priori*, ce qui permet de faire face à l'hétérogénéité des relations rencontrées en domaine ouvert, notamment sur le Web. Le type de ces relations doit alors être découvert de façon automatique à partir des textes. Dans ce cadre, les structures d'information considérées sont fréquemment des relations binaires intervenant entre des entités nommées, à l'instar de Hasegawa et al. (2004). Ce travail, parmi les premiers sur cette problématique, a avancé l'hypothèse que les relations les plus intéressantes entre entités nommées sont aussi les plus fréquentes dans une collection de textes, de sorte que les instances de relations susceptibles de former des clusters de grande taille peuvent être distinguées des autres. Pour opérer cette distinction, un seuil de similarité minimale appliqué à une représentation des relations de type sac de mots était établi pour défavoriser les clusters de petite taille. Des améliorations ont par la suite été apportées à cette approche initiale par l'adoption de patrons pour représenter les relations au sein des clusters Shinyama and Sekine (2006) ou

l'usage d'un algorithme d'ordonnancement de ces patrons pour la sélection de relations candidates Chen et al. (2005).

Des systèmes tels que TEXTRUNNER Banko et al. (2007) ou REVERB Fader et al. (2011) se focalisent quant à eux sur l'extraction de relations à partir de phrases en s'appuyant sur un modèle d'apprentissage statistique pour garantir la validité des relations extraites. Des approches à base de règles Akbik and Broß (2009); Gamallo et al. (2012) ou des modèles génératifs Rink and Harabagiu (2011); Yao et al. (2011) ont également été proposés pour ce faire. Tout en restant pour l'essentiel non supervisées, d'autres approches font appel à un utilisateur pour délimiter un domaine d'extraction de façon peu contrainte. Ainsi, le système *On-Demand Information Extraction* Sekine (2006) initie le processus d'extraction par des requêtes de moteur de recherche.

Une part notable des travaux menés en EI non supervisée se focalisent sur l'extraction des relations. Le problème de leur regroupement a été en revanche moins abordé, en particulier pour rassembler des relations équivalentes mais exprimées de façon différente. Nous présentons dans cette thèse une méthode efficace pour à la fois extraire et regrouper des relations entre entités nommées à une large échelle. L'étape d'extraction se fonde sur l'identification de couples d'entités nommées cooccurrant à un niveau phrastique, combinée à une procédure de filtrage pour éliminer les fausses relations Wang et al. (2011). L'étape de regroupement s'appuie sur deux niveaux de regroupement : un premier niveau de regroupement des relations sur la forme, utilisant une mesure de similarité simple, et un second niveau permettant de rapprocher les premiers clusters obtenus en utilisant une mesure de similarité sémantique plus élaborée Wang et al. (2013). Ces deux niveaux se complètent d'un regroupement réalisé suivant une autre dimension, en l'occurrence de nature thématique.

Nous présentons d'abord une vue d'ensemble de l'approche proposée à la section 2. Les sections 3 et 4 détaillent respectivement les méthodes d'extraction et de regroupement des relations. Enfin, les sections 5 et 6 rendent compte de l'évaluation de cette méthode de regroupement sous plusieurs angles et la mettent en perspective.

2 Vue d'ensemble

Le travail de cette thèse s'inscrit dans un contexte plus large visant à développer un processus d'extraction d'information non supervisée susceptible de répondre à des problématiques de veille telle que $\ll$ suivre tous les événements faisant intervenir les sociétés X et Y $\gg$. À la base de ce processus se trouve une notion de relation reprenant

pour l'essentiel les hypothèses des travaux mentionnés ci-dessus : une relation est définie par la cooccurrence de deux entités nommées dans une phrase. Compte tenu du caractère non supervisé de la démarche, l'idée sous-jacente à ces restrictions est de se focaliser en premier lieu sur des cas simples, autrement dit des relations s'appuyant sur des arguments facilement identifiables dans un espace textuel suffisamment limité pour rendre leur caractérisation synthétique et s'affranchir des problèmes de coréférence au niveau de leurs arguments.

Dans les systèmes d'EI non supervisée, les entités en relation peuvent être des entités nommées Hasegawa et al. (2004) ou, de façon plus ouverte, des syntagmes nominaux Rozenfeld and Feldman (2006b). Les entités nommées permettent en général d'avoir une meilleure séparation des différents types de relations alors que l'utilisation de syntagmes nominaux permet d'avoir un plus grand nombre de candidats. Nous nous intéressons dans notre cas aux relations entre entités nommées, à la fois pour faciliter l'organisation des relations trouvées et parce qu'il s'agit du besoin le plus généralement répandu en contexte applicatif de veille.

Plus formellement, comme illustré par la figure 1[1], les relations extraites des textes, que l'on devrait en toute rigueur appeler instances de relations, même si leur type n'est pas explicitement défini, sont caractérisées par trois grandes catégories d'information permettant tout à la fois de les définir et de fournir les éléments nécessaires à leur regroupement :



Figure 1: Exemple de relation extraite

- un couple d'entités nommées (E1 et E2). Dans les expérimentations menées, nous nous sommes restreints aux entités de type personne (PERS), organisation (ORG) et lieu (LIEU) ;

- une caractérisation linguistique de la relation. Il s'agit de la façon dont la relation est exprimée linguistiquement. Chaque relation étant extraite sur la base de la présence dans une phrase d'un couple d'entités nommées correspondant aux types ci-dessus, sa caractérisation linguistique comporte trois parties :

[1]L'exemple est donné en anglais car nos expérimentations ont été réalisées dans cette langue.

- *Cpre* : la partie de la phrase précédant la première entité (E1) ;
- *Cmid* : la partie de la phrase se situant entre les deux entités ;
- *Cpost* : la partie de phrase suivant la seconde entité (E2).

Le plus souvent *Cmid* exprime la relation proprement dite tandis que *Cpre* et *Cpost* fournissent plutôt des éléments de contexte pouvant être utiles dans la perspective de son regroupement avec d'autres relations.

- un contexte thématique : ce contexte est formé des mots pleins du segment de texte thématiquement homogène environnant l'instance de relation extraite

On notera qu'une telle relation revêt une forme que l'on peut qualifier de semi-structurée dans la mesure où une partie de sa définition – le couple d'entités – renvoie à des éléments d'une ontologie prédéfinie tandis que son autre partie n'apparaît que sous une forme linguistique.

Le processus d'extraction d'information non supervisée défini autour de cette notion de relation s'articule quant à lui de la façon suivante :

1. pré-traitement linguistique des textes ;
2. extraction de relations candidates ;
3. filtrage des relations candidates ;
4. regroupement des relations selon leur similarité.

Les trois premières étapes concernent plus particulièrement le procédure d'extraction des relations, la dernière étape couvrant leur regroupement. Le pré-traitement linguistique des textes permet de mettre en évidence dans les textes les informations nécessaires à la définition des relations. Ce pré-traitement comporte donc une reconnaissance des entités nommées pour les types d'entités visés, une désambiguïsation morpho-syntaxique des mots ainsi que leur normalisation. Ces traitements s'appuient sur les outils d'OpenNLP[2]. En outre, chaque document fait l'objet d'une segmentation thématique linéaire de sorte que chaque instance de relation est associée à un segment thématique à partir duquel est construit son contexte thématique. Cette segmentation est réalisée par l'outil LCseg Galley et al. (2003).

[2] `http://opennlp.sourceforge.net`

3 Extraction de Relations Non Supervisée

Extraction initiale des relations candidates

Lors de l'extraction initiale des relations candidates, les contraintes sont très limitées. Sont ainsi extraites les relations correspondant à tout couple d'entités nommées dont les types correspondent aux types ciblés, avec pour seules restrictions la cooccurrence de ces entités dans une même phrase et la présence d'au moins un verbe entre les deux. Le tableau 1 donne le volume des relations ainsi extraites à partir de la sous-partie du corpus AQUAINT-2 constituée de 18 mois du journal *New York Times*, corpus utilisé pour toutes les expérimentations présentées dans cette thèse.

PERS – PERS	175 802	ORG – PERS	73 895	LIEU – ORG	57 092
PERS – ORG	126 281	ORG – ORG	77 025	LIEU – PERS	78 845
PERS – LIEU	152 514	ORG – LIEU	71 858	LIEU – LIEU	116 092

Table 1: Volumétrie des instances de relations par extraction initiale

Un examen de ces relations candidates montre cependant qu'un nombre très significatif des relations ainsi extraites ne correspondent pas à de véritables relations entre les entités impliquées. Il semble donc que cette stratégie basique d'extraction, qui peut donner des résultats intéressants dans des domaines de spécialité[3], ne soit pas suffisamment sélective en domaine ouvert. Nous avons donc cherché à la compléter par un processus de filtrage spécifique visant à déterminer si deux entités dans une phrase sont ou ne sont pas liées par une relation, sans *a priori* sur la nature de cette relation.

Filtrage heuristique

Dans une perspective exploratoire, nous avons défini un nombre restreint d'heuristiques de filtrage et analysé leur impact. Ces heuristiques sont au nombre de trois :

- la suppression des relations comportant entre leurs deux entités un verbe exprimant un discours rapporté (dans le cas présent, la liste se limite aux verbes *to say* et *to present*). Ceci vise à éviter d'extraire une relation entre les entités *Homgren* et *Allen* dans l'exemple suivant:

  Holmgren said Allen was more involved with the team ...

[3]Le travail rapporté dans Embarek and Ferret (2008) montre que dans le domaine médical, les relations extraites sur la base de cette stratégie sont correctes dans 79% des cas.

- nombre de mots entre les deux entités limité à 10. Au-delà de cette limite empirique, le nombre des relations effectives entre les deux entités devient en effet très faible ;
- limitation à 1 du nombre de verbes entre les deux entités, sauf si ces verbes ont valeur d'auxiliaire (*be*, *have* et *do*).

L'application de ces heuristiques aux relations extraites a globalement pour conséquence de réduire leur volume d'environ 50%. Le tableau 2 illustre plus précisément ce ratio pour chaque catégorie de relations considéré à partir d'un échantillon de 8 000 relations pour chaque type.

Catégories	**filtrées/gardées**	**discours**	**distance**	**1 seul verbe**
LIEU – LIEU	4 287 / 3 713 (46%)	440	3 548	2 763
LIEU – ORG	4 097 / 3 903 (49%)	488	3 224	2 650
LIEU – PERS	4 790 / 3 210 (40%)	1 636	3 352	2 638
ORG – LIEU	4 225 / 3 775 (47%)	643	3 324	2 869
ORG – ORG	4 169 / 3 831 (48%)	627	3 123	2 810
ORG – PERS	4 541 / 3 459 (43%)	1 541	3 155	2 859
PERS – LIEU	4 209 / 3 791 (47%)	905	3 199	2 813
PERS – ORG	3 888 / 4 112 (51%)	952	2 742	2 566
PERS – PERS	4 444 / 3 556 (44%)	1 290	3 109	2 741

Table 2: Effet de l'application des heuristiques de filtrage

Chacune des trois dernières colonnes donne le nombre de relations filtrées par l'heuristique considérée, sachant qu'une relation peut-être filtrée par plusieurs heuristiques. La deuxième colonne fournit quant à elle le nombre de relations filtrées et le nombre de celles qui sont conservées, avec le pourcentage que représentent ces dernières. L'heuristique la plus filtrante est clairement celle de la distance entre entités mais celle limitant le nombre de verbes a également un impact très significatif.

Néanmoins, ces ratios de filtrage doivent être mis en parallèle avec une évaluation de l'efficacité des heuristiques correspondantes en termes de sélection des relations correctes. Pour ce faire, nous avons choisi au hasard 50 instances pour chaque catégorie et nous avons procédé à une annotation manuelle de leur validité. Le tableau 3 donne le résultat de cette évaluation montrant que globalement, le taux de fausses relations parmi les relations filtrées est assez élevé pour toutes les catégories de relations mais que parmi les relations conservées, certains catégories de relations, en particulier toutes les relations ayant un lieu comme première entité nommée, se caractérisent par un taux de fausses relations encore très important.

Catégories	Filtrées		Gardées	
	correctes	**fausses**	**correctes**	**fausses**
LIEU – LIEU	1	49 (98%)	9 (18%)	41
LIEU – ORG	4	46 (92%)	8 (16%)	42
LIEU – PERS	3	47 (94%)	2 (4%)	48
ORG – LIEU	7	43 (86%)	14 (28%)	36
ORG – ORG	6	44 (88%)	20 (40%)	30
ORG – PERS	4	46 (92%)	20 (40%)	30
PERS – LIEU	13	37 (74%)	40 (80%)	10
PERS – ORG	12	38 (76%)	40 (80%)	10
PERS – PERS	5	45 (90%)	14 (28%)	36

Table 3: Évaluation du filtrage par les heuristiques

Ce constat n'est d'ailleurs pas surprenant dans la mesure où la première entité d'une relation occupe souvent un rôle d'agent alors que les lieux apparaissent le plus fréquemment comme des circonstants. Compte tenu de cette observation, nous avons choisi d'écarter systématiquement les relations ayant un lieu comme première entité dans la suite des traitements.

Filtrage par apprentissage

L'évaluation précédente a mis en évidence l'intérêt des heuristiques testées pour écarter les mauvaises relations mais a également montré leur insuffisance pour conserver une proportion significative des relations correctes. Nous avons donc choisi d'adjoindre à ces heuristiques un module de filtrage reposant sur un classifieur statistique décidant si une relation extraite est véritablement sous-tendue par une relation effective entre ses entités. La première tâche pour ce faire a été de construire un corpus de référence en annotant manuellement un ensemble de relations.

Plus précisément, 200 relations ont été sélectionnées au hasard et annotées pour chacune des 6 catégories de relations finalement considérées. L'annotation distinguait les relations correctes, les relations incorrectes du fait d'un problème de reconnaissance des entités nommées et les relations fausses du fait de l'absence de relation effective. Les résultats de cette annotation sont présentés dans le tableau 4.

Les relations incorrectes du fait des entités nommées représentent environ 20% de l'ensemble et ont été laissées de côté pour l'entraînement et le test des classifieurs. Le corpus résultant se compose donc de 964 relations, 531 étant correctes et 433 étant fausses, ce

Catégories	correctes	erreurs EN	fausses
ORG – LIEU	38% (77)	18% (35)	44% (88)
ORG – ORG	39% (78)	14% (28)	47% (94)
ORG – PERS	36% (72)	18% (36)	46% (92)
PERS – LIEU	51% (102)	31% (62)	18% (36)
PERS – ORG	60% (120)	18% (36)	22% (44)
PERS – PERS	41% (82)	20% (39)	40% (79)
Tous	44% (531)	20% (236)	36% (433)

Table 4: Résultat de l'annotation manuelle des relations

qui constitue un ensemble suffisamment équilibré pour ne pas poser de problème spécifique pour l'apprentissage des modèles statistiques.

Plusieurs de ces modèles ont été testés en se concentrant d'abord sur des modèles exploitant un ensemble de caractéristiques locales non structurées. Classiquement, nous avons ainsi entraîné un classifieur bayésien naïf, un classifieur de type maximum d'entropie (MaxEnt), un arbre de décision et un classifieur fondé sur les Machines à Vecteurs de Support (SVM). Pour les trois premiers, nous nous sommes appuyés sur l'implémentation fournie par la boîte à outils MALLET McCallum (2002) tandis que pour le dernier, nous avons eu recours à l'outil SVM^{light} Joachims (1999). Ces différents classifieurs ont été entraînés en utilisant le même ensemble de caractéristiques. Ces caractéristiques reprennent celles utilisées classiquement pour l'extraction de relations, à l'instar de Banko and Etzioni (2008) :

- le type des entités nommées E1 et E1 ;
- la catégorie morpho-syntaxique des mots situés entre les deux entités, avec une caractéristique binaire pour chaque couple (*position dans la séquence*, *catégorie*), ainsi que les bigrammes de catégories morpho-syntaxiques entre E1 et E2, avec une caractéristique binaire pour chaque triplet (*position i*, cat_i, cat_{i+1}) ;
- la catégorie morpho-syntaxique des deux mots précédant E1 et des deux mots suivant E2, à la fois en tant qu'unigrammes et en tant que bigrammes ;
- la séquence des catégories morpho-syntaxiques entre E1 et E2. Chaque séquence possible de 10 catégories est encodée comme une caractéristique binaire ;
- le nombre de mots entre E1 et E2 ;

- le nombre de signes de ponctuation (virgule, guillemet, parenthèse ...) entre E1 et E2.

Compte tenu de la taille relativement réduite du corpus pour chaque catégorie de relations, nous avons choisi d'évaluer ces différents classifieurs en faisant appel à la technique classique de la validation croisée. Le corpus annoté a ainsi été découpé en 10 parties égales, 9 parties étant utilisées pour l'entraînement des classifieurs, la partie restante pour le test, le processus étant mené 10 fois afin que chaque partie serve à la fois pour l'entraînement et le test. Les résultats donnés par le tableau 5 sont des moyennes sur ces 10 itérations pour les mesures standard d'*Exactitude* (accuracy), de *Précision*, *Rappel* et *F1-mesure*.

Modèle	**Exactitude**	**Précision**	**Rappel**	**F1-mesure**
Bayésien naïf	0,637	0,660	0,705	0,682
MaxEnt	0,650	0,665	0,735	0,698
Arbre de décision	0,639	0,640	0,784	0,705
SVM	0,732	0,740	0,798	0,767
Banko and Etzioni (2008)	/	0,883	0,452	0,598

Table 5: Évaluation des classifieurs statistiques

Ce tableau montre en premier lieu que les meilleurs résultats sont obtenus par le classifieur de type SVM, ce qui n'est pas surprenant au vu des travaux réalisés de façon générale sur l'extraction de relations. On notera également un certain équilibre entre la précision et le rappel et ce, pour tous les types de classifieurs. Enfin, ces résultats se comparent favorablement à ceux de Banko and Etzioni (2008) sur le même sujet comme le montre la dernière ligne du tableau 5. Dans ce dernier cas, le profil des résultats est un peu différent puisque la précision est plus forte que la nôtre mais le rappel très largement inférieur. Il faut néanmoins préciser que dans Banko and Etzioni (2008), les relations extraites peuvent faire intervenir des entités plus générales que des entités nommées, ce qui est *a priori* un facteur de difficulté.

Modèle de séquence pour le filtrage par apprentissage

À l'instar de Banko and Etzioni (2008), nous avons également testé un classifieur prenant en compte la notion de séquence en nous appuyant sur les *Champs Conditionnels Aléatoires* (CRF). Dans ce cas, la tâche considérée n'est plus directement une tâche de classification des relations mais prend la forme d'un étiquetage, illustré par la figure 2.

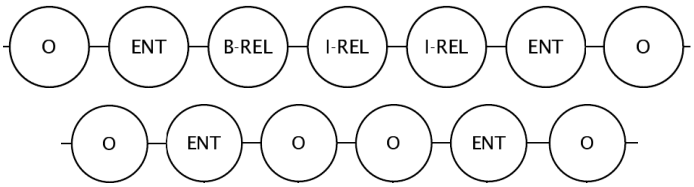


Figure 2: Étiquetage des relations par un modèle CRF

Plus précisément, il s'agit d'étiqueter chaque mot d'une phrase par l'une des quatre étiquettes suivantes, suivant en cela le modèle IOB introduit par Ramshaw and Marcus (1995) :

- O : mot de la phrase en dehors d'une relation ;
- ENT : élément d'une entité nommée définissant une relation potentielle (E1 ou E2) ;
- B-REL : premier mot d'une relation suivant E1 ;
- I-REL : mot faisant partie d'une relation.

Dans ce schéma, une relation est jugée correcte lorsque l'étiquetage suit la première configuration de la figure 2 (avec un nombre de I-REL variable selon la relation) tandis qu'elle est jugée fausse lorsque l'étiquetage produit la seconde configuration [4].

Comme les classifieurs de la section précédente, ce modèle à base de CRF linéaires s'appuie sur un ensemble de caractéristiques :

- la catégorie morpho-syntaxique du mot courant, du mot précédent et du mot suivant ;
- les bigrammes de catégories morpho-syntaxiques $<\text{cat}_{i-1}, \text{cat}_i>$, avec i=-1,0,1 (0 : mot courant ; -1 : mot précédent ; 1 : mot suivant) ;
- le type d'entité nommée du mot courant et de chacun des 6 mots le précédant et le suivant. Ce type peut avoir une valeur NIL lorsque le mot ne fait pas partie d'une entité nommée.

Le tableau 6 montre les résultats obtenus par ce modèle CRF, implémenté au moyen de l'outil Wapiti Lavergne et al. (2010), suivant les mêmes modalités de validation croisée utilisées pour les classifieurs de la section précédente. La comparaison avec le meilleur

[4] D'autres séquences marquant l'absence de relation seraient en principe possibles (comme O – ENT – B-REL – O – O – ENT – O) mais seule la seconde est observée en pratique, sans doute du fait de la présence des deux seuls types de séquences de la figure 2 dans le corpus d'apprentissage.

de ceux-ci met en avant une légère supériorité du modèle à base de CRF, avec toujours le même équilibre entre précision et rappel. C'est donc ce modèle que nous avons retenu pour le filtrage des relations dans le cadre de notre processus d'extraction d'information non supervisée.

Modèle	**Exactitude**	**Précision**	**Rappel**	**F1-mesure**
SVM	0,732	0,740	0,798	0,767
CRF	0,745	0,762	0,782	0,771

Table 6: Évaluation du modèle CRF

Application du filtrage des relations

L'extraction des relations telle que nous l'avons envisagée précédemment se compose des 4 étapes suivantes, appliquées successivement :

1. une extraction initiale ne posant comme contraintes que la cooccurrence dans une phrase d'entités nommées relevant d'un ensemble donné de types et la présence d'au moins un verbe entre les deux ;

2. l'application des heuristiques permettant d'écarter avec une bonne précision un grand nombre de relations fausses ;

3. l'application d'un modèle de filtrage à base de CRF permettant de discriminer plus finement les relations correctes.

4. l'élimination des relations redondantes

Le constat de la présence dans nos relations filtrées d'un certain nombre de relations identiques, pour une part issues d'articles sur un même sujet ou d'articles correspondant à des rubriques très formatées, nous a conduit à compléter le processus de filtrage constitué par les trois premières étapes par un dédoublonnage final visant à éliminer ces relations redondantes. Pour ce faire, nous reprenons les outils utilisés par le processus de regroupement de relations de la section 4 pour évaluer la similarité entre les relations et détecter les relations dont la similarité est maximale ce qui, compte tenu de l'existence d'une borne supérieure pour la mesure utilisée, signifie que les relations sont identiques. Pour chaque ensemble de relations identiques, un représentant est alors choisi. Il est à noter que cette opération de dédoublonnage vient en dernière position à la fois parce que son coût est le

plus important mais également parce qu'elle repose sur l'évaluation de la similarité entre les relations, exploitée ensuite directement pour le regroupement des relations.

	Initial	**Heuristiques**	**Classifieur CRF**	**Dédoublonnage**
ORG-LIEU	71 858	33 505 (47%)	16 700 (23%)	15 226 (21%)
ORG-ORG	77 025	37 061 (48%)	17 025 (22%)	13 704 (18%)
ORG-PERS	73 895	32 033 (43%)	12 098 (16%)	10 054 (14%)
PERS-LIEU	152 514	72 221 (47%)	55 174 (36%)	47 700 (31%)
PERS-ORG	126 281	66 035 (52%)	50 487 (40%)	40 238 (32%)
PERS-PERS	175 802	78 530 (45%)	42 463 (24%)	38 786 (22%)
TOTAL	677 375	319 385 (47%)	193 947 (29%)	165 708 (24%)

Table 7: Niveau de filtrage des relations à l'issue de chacune des étapes

Le tableau 7 illustre l'application des 4 étapes de filtrage aux relations du tableau 1. On constate que ce filtrage laisse de côté un grand nombre des relations extraites initialement mais que le volume des relations restantes est *a priori* suffisant pour alimenter efficacement les étapes suivantes de notre processus d'extraction d'information non supervisée. Par ailleurs, comme Banko and Etzioni (2008), nous nous situons dans un contexte de traitement de volumes textuels importants caractérisés par une certaine redondance informationnelle où la perte d'une certaine quantité d'instances de relations n'est pas un obstacle pour appliquer notre approche.

4 Regroupement de Relations

Principe du regroupement des relations

À l'instar de travaux dans le domaine de l'EI non supervisée comme Shinyama and Sekine (2006) ou Rozenfeld and Feldman (2007), notre objectif final est le regroupement des relations selon leur similarité, en particulier pour en faciliter l'exploration. La méthode de regroupement visée doit à la fois être capable de traiter le volume important de relations issu de leur filtrage et la variété de la forme de ces relations, inhérente au fait de travailler en domaine ouvert. Pour ce faire, nous proposons une méthode, illustrée par la figure 3, s'organisant en deux étapes principales, à l'image de l'approche multi-niveau de Cheu et al. (2004) : un premier clustering de base est réalisé en s'appuyant sur la similarité des formes de surface des relations, ce qui permet de former de manière efficace de petits clusters homogènes en regroupant des instances de relations définies autour d'un même mot-clé principal, comme pour les formes {*create, create the, that create, who create the, etc*}. ;

une seconde étape de clustering est ensuite appliquée pour rassembler ces clusters initiaux sur la base d'une similarité sémantique entre relations plus complexe. Cette similarité permet de prendre en compte des phénomènes tels que la synonymie, voire la paraphrase, pour rassembler des formes telles que {*create, establish, found, launch, inaugurate, etc*}.

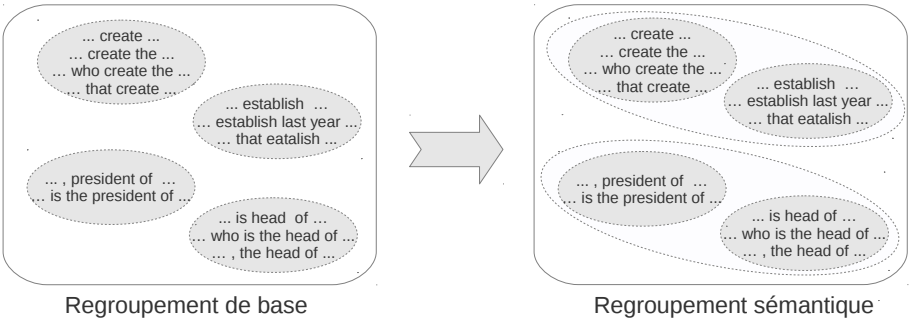


Figure 3: Regroupement des relations en niveaux

Ces deux niveaux de clustering se complètent d'un troisième type de clustering, dont l'objet est différent mais complémentaire des deux premiers : son objectif est de rassembler les instances de relations dont les contextes font référence au même thème. Cette autre dimension de structuration des instances de relations possède à la fois un intérêt applicatif et un intérêt du point de vue sémantique : le fait de se situer dans un contexte thématique homogène tend en effet à réduire le problème de l'ambiguïté sémantique des mots, ce qui permet des rapprochements plus sûrs.

Regroupement de base

En EI non supervisée, le nombre de relations extraites est rapidement important. De ce fait, il est quasiment impossible d'appliquer des mesures de similarité sémantique élaborées entre toutes les relations extraites. Nous mettons en œuvre un premier niveau de clustering afin de former des regroupements de relations proches les unes des autres sur le plan de leur expression linguistique, comme le fait de regrouper *create the* et *who create*. Pour ce faire, nous nous sommes appuyés sur une similarité *Cosinus* appliquée à une représentation de type sac de mots de la partie *Cmid* des relations. Outre son compromis intéressant entre simplicité et efficacité, ce choix a été motivé par la possibilité d'appliquer cette similarité aux larges ensembles de relations extraites dans notre contexte par une utilisation de l'algorithme *All Pairs Similarity Search* (APSS) Bayardo et al. (2007). Moyennant la fixation *a priori* d'un seuil de similarité minimale, celui-ci permet en effet de construire de façon optimisée la matrice de similarité d'un ensemble de vecteurs suivant la mesure *Cos-*

inus. Cette matrice étant calculée et transformée en graphe de similarité, nous appliquons ensuite l'algorithme *Markov Clustering* Van Dongen (2000) pour former les regroupements de relations. Cet algorithme identifie les zones d'un graphe de similarité les plus densément connectées en réalisant des marches aléatoires dans ce graphe. Outre son efficacité, il présente l'avantage, du point de vue de l'IE non supervisée, de ne pas nécessiter la fixation préalable d'un nombre de clusters.

Pondération des termes

Si l'on considère que tous les mots d'une phrase n'apportent pas la même contribution au sens général de la phrase, il est nécessaire d'établir une bonne stratégie de pondération pour établir une bonne mesure de similarité entre phrases. Trois types de pondération sont considérés ici:

- pondération binaire: tous les mots de $Cmid$ ont le même poids (1.0);
- pondération *tf-idf*: un poids *tf-idf* standard est attribué à chaque mot (prenant en compte la fréquence du mot dans la relation et la fréquence inverse du mot dans l'ensemble des relations);
- pondération grammaticale: des poids spécifiques sont donnés aux mots en fonction de leur catégorie morpho-syntaxique.

La pondération binaire est la plus simple et forme une *baseline*, qui a été utilisée dans nos premières expériences, en particulier en raison de l'efficacité de l'implémentation de l'APSS avec un poids binaire. La pondération *tf-idf* prend en compte, par le biais du facteur *idf*, une mesure de l'importance du terme dans le corpus. Néanmoins, la fréquence des mots dans un corpus n'est pas forcément corrélée à leur rôle dans la caractérisation d'une relation. Par exemple, le verbe *buy* peut être fréquent dans un corpus de documents financiers (et donc avoir un poids faible), mais il n'en sera pas moins représentatif de la relation BUY(ORG-ORG). C'est pourquoi nous avons décidé d'introduire une pondération grammaticale.

Une analyse des catégories morpho-syntaxiques nous a amenés à les séparer en plusieurs classes selon leur importance dans la contribution à l'expression d'une relation. Plus précisément, nous considérons quatre classes:

- **(A) contribution directe**, de poids élevé: les mots de cette classe contribuent directement au sens de la relation et incluent les verbes, noms, adjectifs, prépositions;

- **(B) contribution indirecte**, de poids moyen: les mots de la classe B ne sont pas directement liés au sens de la relation mais sont pertinents dans l'expression de la phrase, comme les adverbes et les pronoms;

- **(C) information complémentaire**, de poids faible: cette classe contient des mots fournissant une information complémentaire sur la relation, comme les noms propres;

- **(D) pas d'information**, de poids nul: cette classe contient les mots vides que l'on veut ignorer (symboles, nombres, déterminants etc.).

Nous présentons dans le tableau 8 une configuration de pondération grammaticale. La liste des catégories morpho-syntaxiques est fondée sur les catégories du *Penn Treebank*. Des poids de 1,0, 0,75, 0,5 et 0 sont attribués aux classes A, B, C, D. Pour les catégories non présentes dans cette liste, un poids par défaut de 0,5 est utilisé.

Classe	**Catégories morpho-syntaxiques**
A (w=1,0)	VB VBD VBG VBN VBP VBZ NN NNS JJ JJR JJS IN TO RP
B (w=0,75)	RB RBR RBS WDT WP WP$ WRB PDT POS PRP PRP$
C (w=0,5)	NNP NNPS UH
D (w=0,0)	SYM CC CD DT MD

Table 8: Pondération grammaticale: distribution des poids selon la catégorie morpho-syntaxique

Regroupement par mots-clés représentatifs

Pour renforcer ce premier niveau de clustering, la stratégie généraliste présentée ci-dessus a été complétée par une heuristique tenant compte de la spécificité des relations. Au sein d'un cluster de base, la forme linguistique de ces dernières est en effet souvent dominée par un verbe (*founded* pour *a group founded by* ou *which is founded by*) ou par un nom (*head* pour *who is the head of*, *becomes head of*), ce terme dominant possédant une fréquence élevée dans le cluster. De ce fait, nous considérons le nom ou le verbe le plus fréquent au sein d'un cluster de base comme son représentant, à l'instar de travaux comme Hasegawa et al. (2004), et nous fusionnons les clusters partageant le même terme dominant, appelé *mot-clé* dans ce qui suit, pour former des clusters de base plus larges.

Regroupement sémantique

Le premier niveau de clustering ne peut clairement pas regrouper des relations exprimées avec des termes complètement différents. Dans l'exemple *who create the* et *that establish* présenté à la figure 3, les deux formes linguistiques ont peu en commun. Nous avons donc considéré l'ajout d'un second niveau de clustering ayant pour objectif de regrouper les clusters formés précédemment sur des bases plus sémantiques, plus précisément en intégrant les similarités sémantiques au niveau lexical. Contrairement au premier, ce second niveau bénéficie en outre du fait de travailler à partir de clusters et non de relations individuelles, ce qui permet d'exploiter une information plus riche. Il nécessite de ce fait de définir trois niveaux de similarité sémantique : similarité entre les mots, entre les relations et entre les clusters de base de relations.

Évaluation de la similarité sémantique entre les mots

Les mesures de similarité sémantique au niveau lexical se répartissent en deux grandes catégories aux caractéristiques souvent complémentaires : la première rassemble les mesures fondées sur des connaissances élaborées manuellement prenant typiquement la forme de réseaux lexicaux de type WordNet ; la seconde recouvre les mesures de nature distributionnelle, construites à partir de corpus. Pour évaluer la similarité sémantique entre relations, nous avons choisi de tester des mesures relevant de ces deux catégories afin de juger de leur intérêt respectif.

Concernant le premier type de mesures, le fait de travailler avec des textes en anglais ouvre le champ des différentes mesures définies à partir de WordNet. Nous en avons retenu deux caractéristiques : la mesure de Wu et Palmer Wu and Palmer (1994), qui évalue la proximité de deux synsets en fonction de leur profondeur dans la hiérarchie de WordNet et de la profondeur de leur plus petit ancêtre commun ; la mesure de Lin Lin (1998b), qui associe le même type de critère que la mesure de Wu et Palmer et des informations de fréquence d'usage des synsets dans un corpus de référence. Ces mesures étant définies entre synsets, pour se ramener à une mesure entre mots, nous avons adopté la stratégie utilisée notamment dans Mihalcea et al. (2006) consistant à prendre comme valeur de similarité entre deux mots la plus forte valeur de similarité entre les synsets dont ils font partie.

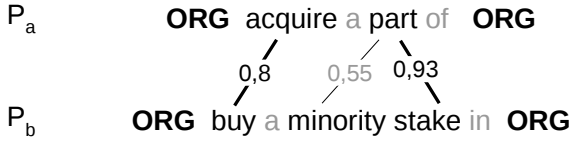
Les mesures de similarité distributionnelles sont quant à elles fondées sur l'hypothèse que les mots apparaissant dans les mêmes contextes tendent à avoir le même sens. La notion de contexte renvoie ici traditionnellement à l'ensemble des mots cooccurrant avec le mot cible dans un corpus. Cette cooccurrence peut être purement graphique, au sein d'une fenêtre de taille fixe, ou bien reposer sur des relations syntaxiques. Nous avons testé ici les

deux types de cooccurrents, les termes au sein des contextes ainsi formés étant pondérés grâce à la mesure d'*Information Mutuelle* et les contextes eux-mêmes étant comparés grâce à la mesure *Cosinus* pour évaluer la similarité de deux mots. Plus précisément, nous avons utilisé les thésaurus distributionnels présentés dans Ferret (2010) pour disposer de ces similarités sous une forme précalculée.

Dans le cadre de la comparaison de relations, nous nous sommes intéressés essentiellement à la similarité sémantique entre des mots appartenant à la même catégorie morpho-syntaxique en nous fondant sur le fait que les relations extraites se définissent généralement autour d'un verbe (*e.g. ORG found by PER*, *ORG establish by PER*) ou d'un nom (e.g.*ORG be partner of ORG*, *ORG have cooperation with ORG*), mais pas sous les deux formes pour un même type de relations, sans doute à cause de la focalisation sur la partie *Cmid* des relations.

Similarité sémantique des relations

La similarité s'applique ici à l'échelle de la définition linguistique des relations, *i.e.* leur partie *Cmid*, ce qui s'apparente à la problématique de la détection de paraphrases. De ce fait, nous avons repris le principe expérimenté dans Mihalcea et al. (2006) pour cette tâche : chaque phrase (ici relation) à comparer est représentée sous la forme d'un sac de mots et lors de l'évaluation de la similarité $sim(P_a, P_b)$ d'une phrase P_b par rapport à une phrase P_a, chaque mot de P_a est apparié au mot de P_b avec lequel sa similarité sémantique, au sens de la section 4, est la plus forte. Ainsi, dans l'exemple ci-dessous, *acquire* est apparié à la seule possibilité, *buy*, tandis que *part* est apparié à *stake*, avec lequel il partage la plus grande similarité selon la mesure de Wu-Palmer.



Un mot d'une phrase peut éventuellement ne pas être apparié si sa similarité avec tous les autres mots de l'autre phrase est nulle. Cette mesure de similarité n'étant pas symétrique, la similarité complète est égale à la moyenne de $sim(P_a, P_b)$ et $sim(P_b, P_a)$. Plus formellement, si l'on définit P_a et P_b comme :

$$P_a = W_1 : f_1,\ W_2 : f_2,\ \ldots, W_i : f_i, \ldots,\ W_M : f_M$$
$$P_b = W_1 : f_1,\ W_2 : f_2,\ \ldots, W_j : f_j, \ldots,\ W_N : f_N$$

où W_k est un mot d'une phrase et f_k, sa fréquence dans la phrase, cette similarité complète s'écrit :

$$S_{P_{a,b}} = \frac{1}{2}\Big(\frac{1}{\sum_{i\in[1,M]} w_i} \sum_{i\in[1,M]} \max_{j\in[1,N]} \{S_{W_{i,j}}\} \cdot w_i + \frac{1}{\sum_{j\in[1,N]} w_j} \sum_{j\in[1,N]} \max_{i\in[1,M]} \{S_{W_{i,j}}\} \cdot w_j\Big) \quad (1)$$

où $S_{W_{i,j}}$ est la similarité sémantique entre les mots W_i et W_j, qu'elle soit fondée sur WordNet ou sur un thésaurus distributionnel et w_i et w_j sont les poids de ces mots respectivement dans P_a et P_b, définis par leur fréquence ($w_i = f_i, w_j = f_j$).

Similarité sémantique des clusters

Le principe adopté pour la similarité de deux relations est trop coûteux à transposer à l'échelle des clusters car il nécessiterait, pour un cluster C_a de cardinalité A et un cluster C_b de cardinalité B, de calculer $A \cdot B$ similarités, lesquelles ne peuvent pas être précalculées comme pour les mots. La similarité à l'échelle des relations étant fondée sur une représentation de type sac de mots, nous avons choisi de construire pour les clusters une représentation de même type, obtenue en fusionnant les représentations de leurs relations. Au sein de la représentation d'un cluster, chaque mot se voit associer sa fréquence parmi les relations du cluster, les mots de plus fortes fréquences étant supposés les plus représentatifs du type de relation sous-jacent au cluster.

Concernant l'évaluation de la similarité entre les clusters, nous avons donc repris la définition de la similarité entre les relations mais avec une légère adaptation destinée à palier le biais pouvant être induit par une trop grande différence d'effectifs entre les deux clusters. Ainsi, dans l'exemple ci-dessous, les clusters C_a et C_b ne sont pas sémantiquement similaires mais leur similarité serait élevée avec une mesure telle que $S_{P_{a-b}}$ du fait du poids élevé du mot *actor* dans C_a. Même si dans un tel cas, $sim(P_b, P_a)$ serait plus faible que $sim(P_a, P_b)$, $sim(P_b, P_a)$ influencerait fortement la moyenne des deux et conduirait à une similarité globale assez forte.

C_a = found:3, actor:3 ... {*i.e.* PER *an actor who found* ORG}
C_b = study:9, actor:1 ... {*i.e.* PER *study at* ORG, PER *an actor study at* ORG}

Pour contrecarrer cet effet, nous introduisons la fréquence des mots dans les deux clusters et non dans celui servant de référence seulement, en remplaçant, dans l'équation (4.19), les poids w_i et w_j par w_{ij}, défini par $w_{ij} = f_i \cdot f_j$.

Regroupement thématique des relations

Les deux niveaux de regroupement de relations (regroupement de base et regroupement sémantique) présentés ci-dessus ont pour objectif de regrouper des instances de relations équivalentes sur le plan sémantique et ce, en s'appuyant uniquement sur des informations locales aux phrases les contenant, en l'occurrence leur partie *Cmid*. Mais chaque relation s'inscrit également dans un contexte plus large, faisant référence à des thèmes tels que la *politique*, l'*économie* ou le *sport* par exemple. En proposant de regrouper les instances de relations suivant cette dimension thématique, nous poursuivons deux objectifs : sur un plan applicatif, proposer une autre dimension de regroupement, complémentaire de la dimension sémantique ; sur le plan du regroupement même des instances de relations, former des clusters sémantiques plus précis en désambiguïsant indirectement les mots des relations sur lesquels ils reposent. Deux instances de relations peuvent en effet avoir été regroupées sur la base d'un mot utilisé avec des sens différents car faisant référence à des contextes thématiques différents, à l'instar par exemple du mot *title* qui possède un sens particulier dans le domaine du sport et un autre dans le domaine des arts.

Ce regroupement thématique est plus précisément effectué de façon indirecte : il ne s'applique pas en premier lieu aux instances de relations mais aux contextes, *i.e.* segments thématiques, dans lesquels elles apparaissent. Tous les segments extraits du corpus considéré sont ainsi regroupés selon leur similarité en adoptant les mêmes modalités que pour le clustering de base des instances de relations : une représentation de type $\ll$ sac de mots $\gg$ pour chaque segment avec une pondération des mots de type *tf.idf*, l'utilisation de la mesure *Cosinus* pour l'évaluation de leur similarité et celle du couple APSS – Markov Clustering pour le regroupement proprement dit. Chaque cluster formé, illustré par les C_i de la figure 4, possède une double représentation : en tant que regroupement de segments thématiques, il incarne un thème du corpus mais chaque segment constituant le contexte d'une ou plusieurs instances de relations, il correspond également à un regroupement d'instances de relations (cf. R_{ij} au sein des C_i) référant au même contexte thématique. Comme l'illustre la figure 4, le regroupement thématique des relations est obtenu par l'intersection des clusters sémantiques et des clusters de segments thématiques : au sein de chaque cluster sémantique, les instances de relations faisant partie d'un même cluster C_i sont regroupées pour former un cluster thématique de relations. Les instances de relations d'un cluster sémantique non regroupées à l'issue de ce processus forment elles-mêmes un cluster thématique.

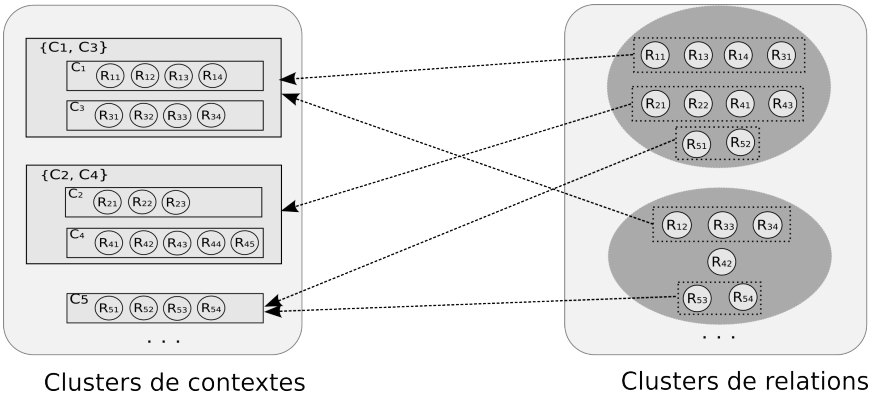


Figure 4: Regroupement thématique des relations

Algorithme de regroupement

Pour la construction de nos clusters de base, nous avons fait appel à l'association d'un seuillage sur les valeurs de similarité entre relations au travers de l'utilisation de l'APSS et de l'algorithme Markov Clustering. Le seuillage réalisé conduit à éclaircir le graphe de similarité et rend possible l'application du Markov Clustering qui, en dépit de son efficacité, ne pourrait gérer la matrice complète de similarité des relations. Par ailleurs, la taille des clusters à former peut être assez variable selon le contenu du corpus considéré mais la valeur de similarité de deux relations est assez facile à étalonner à partir de résultats de référence (cf. section 5 pour une illustration), ce qui justifie le fait de se focaliser sur la similarité entre relations. La problématique est similaire pour le regroupement thématique de relations : la taille des clusters formés peut être assez variable selon le niveau de représentation d'un thème dans le corpus considéré mais la similarité de deux segments thématiques est suffisamment indicative pour fixer des seuils.

Le cas du clustering sémantique est assez différent. Le fait d'utiliser des ressources de natures assez diverses rend difficile la fixation *a priori* d'un seuil de similarité car les intervalles de valeurs ne sont pas les mêmes selon les cas. En revanche, la richesse des ressources sémantiques utilisées permet d'avoir une idée approximative du nombre de voisins d'un cluster de base. Un tel cluster se définissant souvent autour d'un terme clé, ce nombre de voisins est assez directement en rapport avec le nombre de synonymes ou de mots sémantiquement liés à ce terme. De ce fait, pour le clustering sémantique, nous avons adopté l'algorithme *Shared Nearest Neighbor* (SNN) proposé dans Ertöz et al. (2002) plutôt que le Markov Clustering utilisé initialement. Cet algorithme définit en effet implicitement

la taille des clusters qu'il forme en seuillant le nombre de voisins possibles pour chaque élément à regrouper[5].

5 Résultats et Évaluations

Nous avons mené l'évaluation de ce clustering de relations multi-niveau selon une approche externe en utilisant les mesures standard de *précision* et *rappel* (combinés par la *F-mesure*). Ces mesures sont appliquées à des paires de relations en considérant que les relations peuvent être regroupées dans le même cluster ou séparées dans des clusters différents et ce, de façon correcte ou incorrecte par rapport à la référence. Nous utilisons également les mesures standard pour le clustering de *pureté*, *pureté inverse* and *Information Mutuelle Normalisée* (NMI) Amigó et al. (2009). Le clustering de référence utilisé a été construit manuellement à partir d'un sous-ensemble de relations provenant de l'extraction initiale. Il est formé de 80 clusters couvrant 4 420 relations: une douzaine de clusters sont construits pour chaque paire de types d'entités en relation, avec des tailles variant entre 4 et 280 relations. De plus amples détails sur la construction de cette référence et les mesures d'évaluation utilisées sont donnés dans Wang et al. (2012).

Évaluation de l'impact du filtrage sur le regroupement des relations

Nous avons en premier lieu évalué l'impact de la procédure de filtrage sur les résultats du regroupement des relations. Pour ce faire, nous avons appliqué le clustering de base sur l'ensemble des instances de relations extraites avant la procédure de filtrage et sur celui obtenu après le filtrage. Le seuil de similarité utilisé pour ce clustering de base (pour élaguer la matrice de similarité grâce à l'algorithme APSS) a été fixé à 0,45. Ce seuil a été choisi empiriquement en étudiant le comportement de l'algorithme de clustering sur les phrases du corpus *Microsoft Research Paraphrase* Dolan et al. (2004) et couvre les trois quarts des valeurs de similarité de ses phrases en état de paraphrase. Les performances de ces deux applications du clustering de base sont données dans le tableau 9.

Les colonnes de ce tableau correspondent respectivement aux mesures de *Précision*, *Rappel*, *F-mesure*, *Pureté*, *Pureté inverse* et *Information Mutuelle Normalisée*, auxquelles s'ajoutent le nombre de clusters et la taille moyenne des clusters. À l'exception de la *pureté*, toutes ces mesures montrent l'impact positif du filtrage des relations sur leur regroupement.

[5]Les hypothèses faites sur l'adéquation entre le type d'éléments à regrouper et les algorithmes de regroupement ont été confirmé expérimentalement : l'algorithme SNN donne de moins bons résultats que le Markov Clustering pour le premier niveau de clustering mais l'ordre s'inverse pour le clustering sémantique.

	Préc.	Rappel	F-score	Pur.	Pur. inv.	NMI	Nb	Taille
sans filtrage	0,708	0,282	0,403	**0,915**	0,381	0,743	82 338	5,54
avec filtrage	**0,756**	**0,312**	**0,442**	0,902	**0,407**	**0,750**	15 833	**7,50**

Table 9: Impact du filtrage sur les résultats du regroupement des relations

Ce comportement de la *pureté* est d'ailleurs compensé d'une certaine façon par une augmentation plus importante de la *pureté inverse*. En outre, la réduction du bruit au niveau des instances de relations résultant de leur filtrage se manifeste aussi par la tendance à former des clusters plus grands, la taille moyenne de ceux-ci passant de 5,54 à 7,50 instances de relations. Le filtrage favorise donc le rapprochement des instances de relations.

Évaluation du clustering de base

Le même seuil (0,45) présenté à la section précédant est utilisé pour la pondération binaire et celle par tf-idf. Pour la pondération grammaticale, qui est moins stricte, un seuil de 0,60 est utilisé. Les résultats obtenus pour le clustering de base sont présentés dans le tableau 10.

	Préc.	Rappel	F-score	Pur.	Pur. inv.	NMI	Nb	Taille
binaire	0,756	0,312	0,442	0,902	0,407	0,750	15 833	7,50
tf-idf	0,203	**0,445**	0,279	0,646	**0,573**	0,722	11 911	11,44
gramm.	**0,810**	0,402	**0,537**	**0,963**	0,513	**0,812**	13 648	7,56
mots-clés	**0,812**	**0,443**	**0,573**	0,953	0,552	**0,825**	11 726	8,80

Table 10: Résultats du clustering de base pour plusieurs pondérations en utilisant le Markov Clustering (MCL) et un premier regroupement par mots-clés

Le regroupement sur la base de la similarité utilisant une pondération grammaticale donne les meilleurs résultats, avec une meilleure précision et un rappel satisfaisant. Cette pondération utilise en effet plus de connaissances pour mettre en évidence le rôle des verbes, noms ou adjectifs et diminuer l'influence des mots vides qui ne contribuent qu'à des variations linguistiques légères (*who* + verbe, *the one that* + verbe). La pondération *tf-idf* donne quant à elle de moins bons résultats. Cette pondération favorise en effet les mots rares. Or, les noms communs et les verbes, qui supportent le plus souvent les relations, sont plus fréquents que des noms propres ou des occurrences de nombres, par exemple, qui se verront attribuer un score important avec cette pondération alors qu'ils n'apportent pas d'information sur la relation.

Les résultats utilisés par la suite pour le clustering sémantique sont ceux obtenus avec la pondération grammaticale[6], sur laquelle l'étape de regroupement par mots-clés amène une amélioration légère de la F-mesure, due à un accroissement du rappel; mais cette étape permet surtout de réduire le nombre de clusters et d'augmenter leur taille moyenne, comme illustré par les deux dernières colonnes du tableau 10.

Évaluation du clustering sémantique

Pour évaluer l'amélioration apportée par le clustering sémantique, nous comparons les approches proposées à un clustering idéal (*idéal*) donnant le meilleur regroupement possible des clusters de base obtenus par la première étape: chaque cluster de base est associé au cluster de référence avec lequel il partage le plus de relations; puis les clusters associés aux mêmes clusters de référence sont regroupés.

En pratique, pour les mesures fondées sur WordNet, la mesure de Wu-Palmer donne de bons résultats pour les similarités entre noms alors que la mesure de Lin donne de meilleurs résultats pour les verbes. La première est calculée grâce à NLTK (`nltk.org`) tandis que pour la seconde, nous utilisons les similarités précalculées entre les verbes de WordNet de Pedersen (2010). Les similarités distributionnelles sont quant à elles évaluées à partir du corpus AQUAINT-2, sur la base d'une mesure *Cosinus* entre des vecteurs de contexte obtenus soit avec une fenêtre glissante de taille 3 (Dist_{cooc}), soit en suivant les liens syntaxiques entre les mots (Dist_{syn}). Pour l'algorithme SNN, le voisinage de chaque instance de relation est limité aux 100 plus proches relations. Les résultats obtenus sont présentés dans le tableau 11.

	Préc.	**Rappel**	**F-score**	**Pur.**	**Pur. inv.**	**NMI**	**Nb**	**Taille**
WordNet	0,821	0,507	0,627	0,942	0,622	0,839	9 403	10,98
Dist_{cooc}	0,814	0,540	0,649	0,932	0,634	0,841	10 161	10,16
Dist_{syn}	**0,831**	**0,549**	**0,661**	**0,950**	**0,645**	**0,847**	10 116	10,20
idéal	0,847	0,788	0,816	0,957	0,831	0,899	13 468	7,66

Table 11: Résultats du clustering sémantique

La similarité distributionnelle syntaxique donne les meilleurs résultats, bien que comparables à ceux de la similarité distributionnelle graphique. Les deux approches distributionnelles sont meilleures pour cette tâche que celle fondée sur WordNet, ce qui signifie que la méthode pourra plus facilement être adaptée à d'autres langues. Comparés au clustering

[6] Plusieurs seuils et configurations de pondérations grammaticales ont été testés. La version présentée (seuil de 0,60 et poids du tableau 8) est celle donnant les meilleurs résultats.

de base, toutes les méthodes de clustering sémantique montrent une augmentation notable pour toutes les mesures (le F-score passe de 57,3% à 77,3%).

Pour les similarités WordNet, d'autres tests ont été effectués pour vérifier l'importance relative des différentes catégories grammaticales dans ce regroupement. Par exemple, si l'on ne considère que les verbes, les résultats sont un peu inférieurs, en particulier en termes de rappel. Nous avons également expérimenté l'intégration des adjectifs dans la mesure de similarité, mais les résultats ont montré que ces mots n'ont pas d'influence notable sur le regroupement des relations. D'autres tests intégrant des mesures de similarités entre mots de catégories grammaticales différentes ont été effectués, sans apporter d'amélioration.

Exemples de clusters sémantiques Pour donner une idée qualitative des résultats du clustering sémantique, nous présentons quelques exemples de clusters sémantiques, créés en utilisant la mesure Dist_{cooc}. Un exemple de cluster sémantique obtenu pour chaque type de relation est présenté dans le tableau 12, où chaque mot représente un cluster. Il est clair avec ces exemples que des mots différents mais sémantiquement similaires sont regroupés. Néanmoins, des erreurs subsistent: le fait de ne pas différencier les voies active et passive conduit ainsi à certaines erreurs de regroupement pour les relations entre des entités de même type (par exemple, *purchase* et *be purchased by* pour des relations ORG – ORG).

Catégories	**Clusters sémantiques**
ORG – ORG	purchase, buy, acquire, trade, own, be purchased by
ORG – LOC	start in, inaugurate service to, open in, initiate flights to
ORG – PER	sign, hire, employ, interview, rehire, receive, affiliate
PER – ORG	take over, take control of
PER – LOC	grab gold in, win the race at, reign
PER – PER	win over, defeat, beat, oust, topple, defend

Table 12: Exemples de mots regroupés dans les clusters sémantiques

Évaluation du clustering thématique de relations

Dans le cas du clustering des contextes thématiques des relations, l'algorithme MCL a été appliqué avec un seuil empirique pour la mesure *Cosinus* égal à 0,15. Pour l'évaluation du regroupement thématique des relations, une référence spécifique a été construite en se focalisant sur un cluster sémantique et en répartissant ses instances de relations en fonction des différents thèmes caractérisant leur contexte d'occurrence. Cette référence a ainsi permis de juger de façon précise de l'impact de la structuration thématique du contenu des clusters sémantiques opérée par le clustering thématique. En pratique, nous avons annoté 65 instances de la relation *lead by* pour le type ORG-PER. Ces instances ont été réparties

manuellement en trois sous-groupes correspondant aux trois grands thèmes dans le contexte desquels elles apparaissaient : politique (30 instances), économie (21 instances) et sport (14 instances). L'évaluation par rapport à cette référence du résultat de l'application de la procédure de regroupement thématique au cluster sémantique *lead by* est donnée par le tableau 13.

	Préc.	**Rappel**	**F-score**
sémantique	0.362	**0.842**	**0.507**
thématique	**0.400**	0.219	0.283

Pur.	**Pur. inv.**	**NMI**
0.477	**0.908**	0.127
0.723	0.431	**0.348**

Table 13: Résultats du clustering thématique de relations

Ce tableau fait nettement apparaître une amélioration de la précision du regroupement des instances de relations, accompagnée d'une chute du rappel. La pureté, qui mesure la précision au niveau des clusters, est quant à elle significativement améliorée (passant de 0,477 à 0,723), de même que la mesure NMI (de 0,127 à 0,348). Cette amélioration globale des mesures de précision tend à confirmer l'intérêt de l'utilisation de l'information thématique pour invalider certains rapprochements opérés sur la base de sens différents de certains mots. Parallèlement, la chute des mesures de rappel suggère néanmoins que les clusters thématiques formés sont trop petits et opèrent certains distinguos trop spécifiques. Nous avons vérifié ce dernier point de manière plus qualitative en examinant comment les trois clusters thématiques de notre référence se répartissaient parmi les clusters formés par notre procédure de regroupement thématique. Le tableau 14 donne pour chaque cluster de référence quelques uns de ces clusters formés, caractérisés par leurs mots les plus fréquents.

Ce tableau montre clairement que chaque grand thème se retrouve divisé en plusieurs sous-thèmes. Ainsi, un thème comme *Sport* se retrouve en pratique éclaté en sous-thèmes renvoyant à des sports particuliers, comme le baseball, le basketball ou la boxe. La présence de mots partagés entre ces différents sports comme *game*, *play* ou *season* ne suffit pas en effet à les rassembler. De ce point de vue, on peut noter en particulier l'influence du nom des joueurs ou des équipes, comme les *Sox* et les *Yankee* pour le baseball, les *Lakers* et *Bryant* pour le basketball ou *Ruiz* et *Toney* pour la boxe entre autres. Les différents sports impliquent également des actions et donc des verbes particuliers comme *hit* et *pitch* pour le baseball, *shot* pour le football et le basketball ou *fight* pour la boxe. L'ajout d'une information thématique permet donc de différencier ces différents sports mais la structurer de manière plus hiérarchique conduirait à ne pas perdre la capacité à opérer des regroupements plus larges de relations.

Thème	Mots caractéristiques
Politique	
1	**iraq american** official baghdad sunni **force military** kill bush **government** police
2	**oil** price **energy** company **gas** state gasoline production bill saudi barrels **government**
3	**palestinian israel** gaza sharon **hamas** bank minister state **government security**
Économique	
1	share **company** quarter oracle **earnings revenue** analyst report rise sales **stock** profit business
2	**china japan trade company american** government world taiwan beijing market dollar **export**
3	cell cancer **research human disease patient** study university breast **treatment** drug **medical health**
Sports	
1	sox **game** yankee red **team season** run series play hit boston win pitch angel start world **league** manager **player**
2	Bryant Lakers **game play** point **season** team O'Neal Odom jackson Kobe player coach quarter shot **NBA**
3	stone Ruiz Toney **fight show** jagger world rolling **play win** title **champion** game heavyweight **boxing**

Table 14: Mots caractéristiques des clusters thématiques au niveau de la référence pour le type de relation *lead_by*

Étude des avantages du clustering multi-niveau

Comme indiqué au début de la section 4, le calcul des similarités sémantiques est beaucoup plus coûteux que le calcul d'une simple mesure *Cosinus*. Le nombre total de relations atteint 165 708 (cf. tableau 7), alors que le nombre de clusters de base n'est que de 11 726 (cf. tableau 10). Un premier avantage du clustering multi-niveau est donc d'éviter de calculer un trop grand nombre de similarités coûteuses. Mais, parallèlement, il permet également d'améliorer la qualité de l'organisation sémantique des relations, en exploitant la redondance d'information présente dans les clusters de base. Pour vérifier cette hypothèse, nous avons comparé, en nous appuyant sur notre référence, la distribution des similarités entre les relations initiales et entre les clusters de base. Dans un premier temps, nous avons examiné toutes les similarités entre deux instances de relations appartenant au même cluster de référence (distribution intra-cluster D_{intra}) et les similarités entre deux instances appartenant à des clusters différents (distribution intra-cluster D_{inter}), avec l'hypothèse que ces distributions sont bien séparées (avec une moyenne élevée pour D_{intra} et basse pour D_{inter}). Dans un second temps, nous établissons les mêmes distributions de similarités pour les clusters de base, en associant à chaque cluster de référence l'ensemble des clusters

de base qu'il recouvre. Les distributions de similarité obtenues sont présentées à la figure 5 pour la similarité Dist_{cooc}, la même tendance étant observée pour les autres similarités.

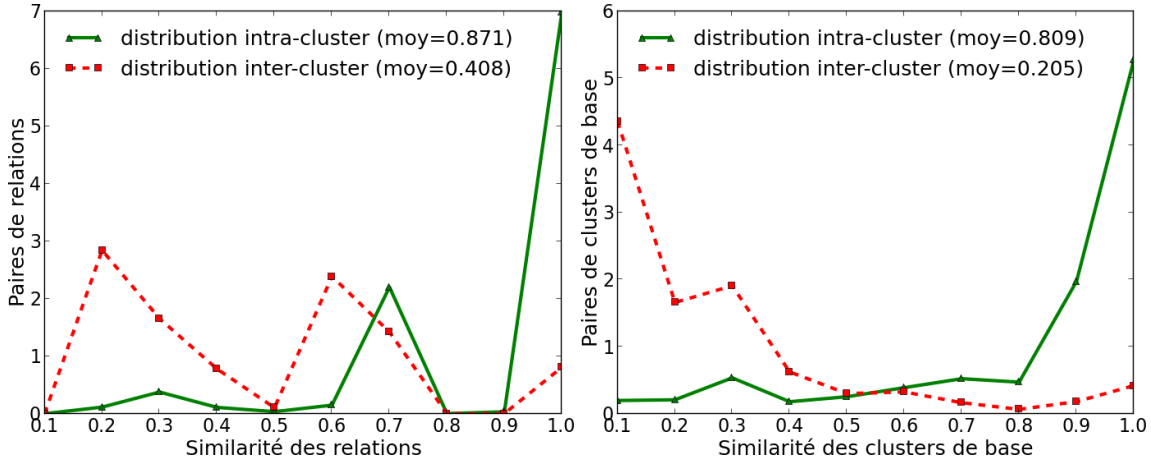


Figure 5: Distribution des similarités entre les relations et entre les clusters de base

On voit clairement sur ces figures que le clustering sémantique effectué à partir des clusters de base peut obtenir de meilleurs résultats parce que les distributions de similarité à l'intérieur des clusters de référence ou entre clusters sont mieux séparées et que la moyenne des similarités pour des relations entre des clusters différents est relativement basse. Ceci confirme notre hypothèse que l'information redondante dans les clusters de base peut être utilisée pour diminuer le bruit causé par les mots non représentatifs de la relation.

6 Travaux liés au clustering de relations

Le clustering de relations occupe des positions diverses dans le domaine de l'EI non supervisée. En premier lieu, il est absent des travaux se concentrant essentiellement sur la découverte et l'extraction de relations, à l'instar du système TEXTRUNNER dans lequel les relations extraites sont directement indexées pour être interrogées. Dans la plupart des autres travaux, la finalité du clustering de relations peut être qualifiée de sémantique dans la mesure où son objectif est de regrouper des relations équivalentes, cette équivalence étant située plus ou moins explicitement sur le plan sémantique. Enfin, quelques travaux plus marginaux, à l'image de Sekine (2006), intègrent également une dimension plus thématique dans les regroupements réalisés.

Même lorsque le clustering de relations possède une vocation sémantique, les moyens pour le mettre en œuvre ne sont pas nécessairement eux-mêmes sémantiques. À l'image de notre premier niveau de clustering, Hasegawa et al. (2004) retrouve ainsi des variations

sémantiques comme (*offer to buy – acquisition of*) au sein des clusters de relations entre entités nommées qu'il forme en appliquant une simple mesure *Cosinus* au contexte immédiat de ces relations. Sekine (2006) va quant à lui un peu plus loin en exploitant un ensemble de paraphrases constitué *a priori* sur la base de cooccurrences d'entités nommées pour faciliter l'appariement de phrases issues de plusieurs articles journalistiques relatant un même événement. Concernant toujours l'évaluation de la similarité entre les relations, Eichler et al. (2008) s'appuie pour sa part sur WordNet pour détecter les relations de synonymie entre verbes. La démarche se rapproche d'une partie de ce que nous avons expérimenté, même si nous avons également inclus les noms dans notre champ d'étude, car ceux-ci sont dominants pour exprimer certaines relations, que nous avons appliqué cette recherche au niveau des clusters de base, et non des relations individuelles, et qu'avec les similarités distributionnelles, nous ne sommes pas restreints aux seules relations de synonymie.

La notion de clustering multiple apparaît quant à elle dans quelques travaux. Kok and Domingos (2008) propose ainsi de construire un réseau de relations sémantiques de haut niveau à partir des résultats du système TEXTRUNNER grâce à une méthode de co-clustering engendrant simultanément des classes d'arguments et des classes de relations. Min et al. (2012) fait quant à lui apparaître deux niveaux de clustering mais avec une optique plus proche de Kok and Domingos (2008) que de la nôtre. Son premier niveau de clustering porte en effet sur les arguments des relations tandis que le second se focalise sur les relations proprement dites. L'objectif du premier niveau de clustering est ainsi de regrouper des relations ayant la même expression et de trouver des arguments équivalents tandis que le second niveau de clustering vise à regrouper des relations ayant des expressions similaires en s'appuyant notamment sur les classes d'arguments dégagées par le premier clustering. Ce dernier exploite un vaste graphe de relations de similarité et d'hyperonymie entre entités construit automatiquement à la fois sur la base de similarités distributionnelles et de patrons lexico-syntaxiques. S'y ajoute pour le second niveau de clustering une large base de paraphrases elle aussi construite automatiquement à partir de corpus.

7 Conclusion et perspectives

Dans cette thèse, nous avons présenté un travail sur l'extraction d'information non supervisée. Nous cherchons d'abord à déterminer si deux entités nommées apparaissant dans une même phrase sont en relation, sans *a priori* sur la nature de cette relation. Nous avons développé pour ce faire une procédure de filtrage par la combinaison d'heuristiques, pour éliminer les cas les plus simples, et d'un classifieur appris à partir d'exemples. Concer-

nant ce dernier, les meilleures performances obtenues par CRF, équilibrées en termes de précision et de rappel, se comparent favorablement aux résultats de Banko and Etzioni (2008), qui ne se limitent cependant pas aux entités nommées comme nous le faisons.

Nous avons présenté également dans cette thèse une méthode de clustering à plusieurs niveaux pour regrouper des relations extraites dans un contexte d'EI non supervisée. Une première étape est appliquée pour regrouper des relations ayant des expressions linguistiques proches de façon efficace et avec une bonne précision. Une seconde étape permet d'améliorer ce premier regroupement en utilisant des mesures de similarité sémantique plus riches afin de rassembler les clusters déjà formés et augmenter le rappel. Nos expériences montrent que dans ce contexte, des mesures de similarité distributionnelle donnent des résultats plus stables que des mesures fondées sur WordNet. Une analyse des distributions des similarités entre les relations initiales et entre les clusters de premier niveau met également en évidence l'intérêt d'un clustering à deux niveaux. Nous avons montré enfin que ce dernier peut être complété par un clustering de nature thématique, apportant à la fois un axe de structuration différent et une amélioration de la précision.

Cette dernière se faisant néanmoins au prix d'une chute du rappel encore trop importante, des travaux complémentaires restent à mener concernant l'intégration des regroupements sémantique et thématique, notamment en considérant un clustering à plus gros grain des contextes thématiques des relations. Par ailleurs, la similarité sémantique des relations pourraient bénéficier de façon plus avancée des travaux menés sur l'identification des paraphrases, en intégrant notamment un ensemble plus large de critères. Enfin, le contexte applicatif de ce travail étant la veille, une évaluation utilisateur reste à mener de ce point de vue, évaluation qui pourrait se faire au travers d'un moteur de recherche sémantique orienté relation dont un prototype a déjà été développé.

Chapter 1

Introduction

Information Extraction (IE) is the task of automatically extracting information from text. The traditional paradigms of this field, which were initially proposed in the series of *Message Understanding Conferences* (Grishman and Sundheim, 1996), are typically represented by tasks such as the filling of predefined templates. These predefined templates are often the obstacle for applying these systems to large corpus in open domain, where the information structures are very heterogeneous. In this thesis, we are interested in one particular and useful information structure: relations between two arguments. For example, in sentence:

George Herbert Walker Bush is the father of *George Walker Bush*.

such *binary relation BeFatherOf* exists between George Herbert Walker Bush and George Walker Bush. Traditional relation extraction systems are often designed for the extraction of relation instances with pre-defined relation types (e.g. *BeFatherOf*). These relation types fixed in advance prevent the system from discovering diverse unknown relation types from a corpus.

New paradigms such as unsupervised IE have gained more and more importance in the last years by relaxing the constraints imposed by predefined information structures or relation types as in traditional IE. This thesis takes place in the context of unsupervised information extraction with the objective to deal with large scale data sets, such as the Web. More precisely, we are interested in the relations between named entities, for addressing issues such as technology watch, for example:

"tracking all events involving companies *X* and *Y*".

This task requires finding relations between company *X* and company *Y* without a specific relation type. If we submit to Google's Web search engine a query such as "Google + * +

Youtube", we can obtain 25 billion webpages, which may contain various unknown relation types. Indeed, given a newspaper corpus or an ensemble of documents retrieved from the Internet, we can not completely pre-define all kinds of relation types for two given named entities, such as "Google" and "Youtube". Therefore, the task of Unsupervised Relation Extraction is to discover different relation types and to extract their relation instances[1].

As one can imagine in the case of the previous query, not all the web pages among the 25 billion ones contain an instance of a reliable relation between these two named entities. Moreover, we are interested by relations that are expressed within a single sentence. Hence, we concentrate more specifically on how instances of these relations that occur at the sentence level can be extracted. Therefore, our first problem to tackle is: *Relation Extraction*. We need to select sentences in which entities *X* and *Y* co-occur and then ensure that a valid relation between these two entities is explicitly expressed in these sentences.

In traditional relation extraction systems, relation instances are extracted for each predefined relation type, so that we know which relation type each extracted instance belongs to. However, in the case of unsupervised relation extraction, we try to make sure the existence of a valid relation between two entities, without necessarily knowing its semantic meaning. One way of characterizing these extracted instances is to group them into clusters according to their similarities. Relation instances can be thus characterized by the groups to which they belong to. Moreover, relation instances are easier to understand for end-users once they are organized in an extensional way. This is the second considered problem in this thesis: *Relation Organization*

Globally speaking, the system proposed in this thesis takes a large amount of raw texts as input and generates an ensemble of clusters of relation instances in two steps: *Relation Extraction* and *Relation Organization*, as illustrated in Figure 1.1.

Relation Extraction: Problems and Solution

One challenge of Unsupervised Relation Extraction is the selection procedure of reliable candidates of relation instances from text without any knowledge about their type. As illustrated by the following examples of named entity pairs co-occurring in sentences, there may be no direct relation between entities (italic words) in such pairs.

- Superintendent *Ed Richard* applauded the "tremendous team effort of workaholics" and *Davis*' tenacious resolve as the key ingredients for the school's success.

[1]The term "Unsupervised" in the task of *Unsupervised Relation Extraction* implies that there is no supervision on the relation types in this task, so that these relation types should be discovered rather than pre-defined.

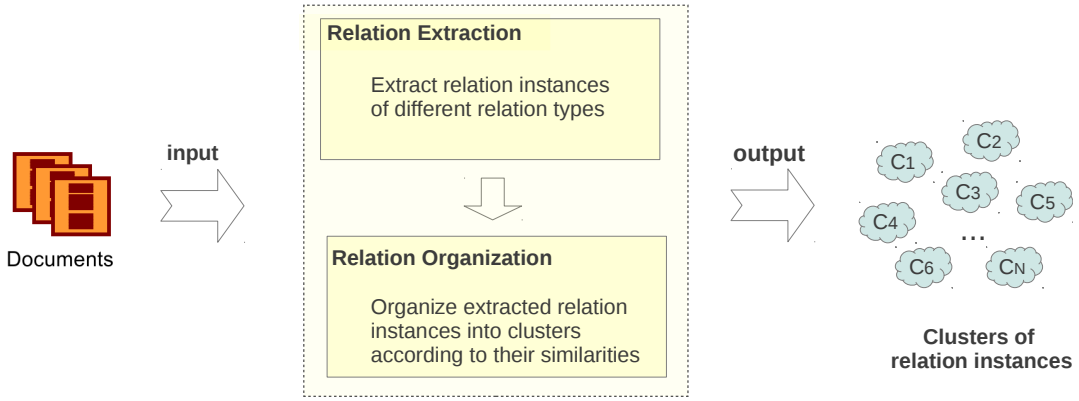


Figure 1.1: Overview of the thesis

- But if *Kerry* wins, the *Republican Party* will all but collapse.

- Among the three identified in previous news reports is one owned by a company apparently set up by the *CIA*, according to *The Washington Post*.

- *Lilly* said that the request had come from *Jim Ellis*, director of DeLay's Americans for a Republican Majority.

For selecting valid relation instances, early approaches started with clustering methods (Hasegawa et al., 2004; Rozenfeld and Feldman, 2007) to group pairs of named entities or noun phrases into clusters according to their similarities. Then, reliable candidates were distinguished by a specific score calculated from the frequency of the pairs, the cluster size or other confidence measures. This kind of approach has at least two limits. On the one hand, the clustering algorithm can be quite time-consuming for massive corpus, which is often the case for unsupervised IE in open domain. As a consequence, the scalability of such relation extraction procedure is limited. On the other hand, the clustering algorithm requires at least a certain quantity of relation instances. It is therefore impossible to extract relation instances from a small size corpus, such as a single document containing only one short review article. More recent approaches train classifiers to extract relation instances by relying only on intra-sentential features (Banko and Etzioni, 2008; Fader et al., 2011). These classifiers are designed to be independent of relation types so that they are applicable for unsupervised relation extraction tasks.

This thesis aims at extracting relation instances efficiently at a large scale. We first propose a model of representation for binary relations, based on which candidates can be initially extracted from corpus. Two steps of filtering are then applied to these initially

extracted candidates to determine their validity as relations. In the first step, a restricted set of heuristics is used to discard efficiently a large number of false relations. The second step relies on a supervised statistical model for refining the selection of candidates.

Relation Organization: Problems and Solution

Relation Organization in unsupervised IE has not often been handled in existing researches yet since most of the work in the field concentrates on the extraction of relation instances. Unsupervised relation organization has to face several difficulties at the same time: massive quantities of extracted relation instances, a great diversity of relation types in open domain and a large set of linguistic variations for relation expression (e.g. synonym, polysemy and paraphrase). The following sentences show for examples that the same relation "acquisition" between two companies can be expressed by several forms, such as "purchase another affiliate", "complete purchase of" or "which acquire":

- *Sprint* purchased another affiliate, *US Unwired*, for 1.3 billion.
- *Sprint Corp.* completes purchase of *US Unwired*.
- *IBM*, which acquired *Lotus* in 1995, said unmarried gay employees should not be surprised by a decision to end domestic-partner benefits, effective January 2006.

On the other hand, the verb "acquire", which refers to a "purchase" event in one of the previous examples, has also other meanings in different contexts:

- *Howard* acquired his political philosophy from *Sir Robert Menzies*.
- *Lee MacPhail* acquired *McKinney* for Stan Bahnsen, who had been the AL rookie of the year in 1968.

In this thesis, we propose a multi-level relation clustering procedure with the purpose of grouping semantically equivalent relations in two steps. The first step, called basic clustering, groups relation instances with similar linguistic expressions to form basic clusters with high precision. This basic clustering is applied with simple similarity measures (e.g. Cosine) calculated on a bag-of-word representation of relation instances so that it can be efficient at a large scale. The second step is a semantic clustering that groups basic clusters into larger semantic clusters using different semantic similarities to handle more complex linguistic phenomena such as synonymy and paraphrase. Semantic similarities at word level, relation instance level and basic cluster level are respectively discussed. Although

these semantic similarities are much more time-consuming compared to simple similarity measures, our multi-level clustering procedure reduces the number of similarities to evaluate from all pairs of relation instances to all pairs of basic clusters, which makes it suitable for this task.

In addition, a topic-based relation clustering is proposed to take into account thematic information for the organization of relation instances. This clustering is first based on the grouping of the contexts of relation instances to form context clusters referring to specific themes. Then, different strategies are investigated for integrating relation clusters and context clusters. Globally, this thematic information is useful both to form more precise relation clusters and to handle polysemy issues.

Clustering Evaluation for Unsupervised IE: Problems and Solution

The organization of relation instances implies several steps of clustering. However, evaluating clustering results is still a difficult issue in general, especially at a large scale, and more particularly for unsupervised IE tasks since references do not exist in this field. In this thesis, we first analyze how internal measures can be applied to evaluate the quality of the clusters of relation instances. Then, an external evaluation approach is investigated by first proposing an interactive way of building a reference. More precisely, extracted relation instances are annotated into clusters according to their relation types so that a cluster reference for a given corpus can be built in a short time. Relying on this manually annotated reference, external measures are finally applied for the evaluation of basic clustering, semantic clustering and topic-based relation clustering.

Organization of the Thesis

The thesis is organized in four main chapters. Chapter 2 presents the state of art concerning different IE tasks and methods, with an emphasis on unsupervised IE. At the end of this chapter, an overview of the system proposed in this thesis is presented. Chapter 3 gives details about how relation candidates are extracted and how candidates corresponding to false relations are filtered out in two steps. Chapter 4 concentrates on the clustering of relation instances. At last, results and evaluations are detailed in Chapter 5.

Chapter 2

State of The Art

The definition of the various tasks of Information Extraction differs according to the different evaluation frameworks that have structured the domain, such as MUC, ACE, etc. An overview of these tasks is presented in the beginning of this chapter. The general tendency of these tasks is to involve less repeated human labor and to allow more flexible information types. IE tasks and designed systems tend to evolve from supervised ones to semi-supervised ones, and then to unsupervised ones. Since our interest in this thesis lies in unsupervised IE, more emphasis will be put on the presentation and comparison of systems for unsupervised IE tasks. Most of the researches represent relations as binary relation triples, applying clustering-based methods, trained classifiers, rule-based models, generative models, etc. Some recent researches adopt N-ary format or automatically constructed templates for relation representations. Different methods and different relation paradigms will be presented and compared. At last, an overview of our proposed unsupervised IE system is given in the end of this chapter.

Contents

2.1 Overview of Information Extraction Evaluations 8

2.2 Supervised Information Extraction 13

2.3 Semi-supervised Information Extraction 14

2.4 Unsupervised Information Extraction 19

2.5 Overview of Our Unsupervised IE System 35

2.1 Overview of Information Extraction Evaluations

In the 1950s, many researches aimed at creating global natural language comprehension systems. Without significant success of this ambition, some researchers oriented their work to the tasks of information extraction. *Message Understanding Conferences* (MUC) are one of the pioneer series of evaluations, focusing on the extraction of local information in sentences since the end of 1980s in United States. Other evaluations concerning information extraction include also *Automatic Content Extraction* (ACE), *Text REtrieval Conference* (TREC), *Text Analysis Conference* (TAC), etc. Most of these evaluation campaigns are in open domain while some others concentrate on specific domains such as biomedical domains (e.g. BioCreAtIvE, i2b2).

2.1.1 MUC Series

There are seven MUC campaigns between 1987 and 1997. The main Information Extraction task in early MUC series is to fill predefined templates with a number of attributes from a text in natural language, mainly from newspapers. An example of such task is shown in Figure 2.1, taken from the MUC-4 (MUC, 1992).

Text :
Salvadoran president-elect Alfredo Cristiani condemned the terrorist killing of Attorney general Roberto Garcia Alvarado and accused The Farabundo Marti National Liberation Front (FMLN) of the crime.

Template :
Location : El Salvador
Incident category : Terrorist Act
Organization : The Farabundo Marti National Liberation Front
Victim : Roberto Garcia Alvarado

Figure 2.1: Template for an attack event from MUC-4

This example refers to an attack event. With the predefined template, it is possible to obtain information linked to this attack event, such as its *location*, *category*, the *organization* which has committed this crime, etc. Similarly, different templates can be defined for other types of events appearing in news reports.

Since the proceeding of MUC-6 in 1995, the *Template Filling* task has started the tendency of designing simpler templates than before for a wide variety of event types, defining attributes for target objects, such as *Organization, Person, and Artifact*. Information about

a specific class of events is extracted and then used to fill the template for each instance of this event (MUC, 1995; Grishman and Sundheim, 1996). In MUC-7 in 1998, the last conference of MUC series, has emerged the task of *Template Relation* (TR), concentrating more on relations independent of the events' scenario, rather than being embedded in templates as attributes or objects. MUC-7 focuses especially on three predefined relations, *location_of, employee_of, product_of* to obtain general relational objects for *Template Element* objects (MUC, 1998).

2.1.2 ACE Series

Following the MUC series, the ACE series were held between 1999 and 2008, involving the detection of entities, relations and events. In these campaigns, entities are first identified in a dedicated *Entity Detection* task, then a *Relation Detection and Characterization* task (RDC) is designed to identify relations between the entities detected. Comparing to the *Template Relation* task of MUC-7, which searches general objects for target *Template Element* objects, the objective of RDC is to check all entity pairs for valid relations of predefined types. In addition, RDC task contains much more expressive relation types than MUC-7. As detailed in Figure 2.2, five general relation are defined, including the role of a person in an organization, part-whole relationships, location relationships, nearby locations, and social relationships. Some of these five types are further sub-divided, which generate 24 types of relations in total (Doddington et al., 2004).

Role: role a person plays in an organization, subtyped as Management, General-Staff, Member, Owner, Founder, Client, Affiliate-Partner, Citizen-Of, or Other
Part: part-whole relationships, subtyped as Subsidiary, Part-Of, or Other
At: location relationships, subtyped Located, Based-In, or Residence
Near: relative locations
Social: subtyped as Parent, Sibling, Spouse, Grandparent, Other-Relative, Other-Personal, Associate, or Other-Professional

Figure 2.2: Relation types defined in ACE (Doddington et al., 2004)

2.1.3 TAC KBP Series

The TAC series started from 2009. One of its main tracks, *Knowledge Base Population* (KBP), encourages researches of automatic systems that discover relational information about named entities and then incorporate this information into a knowledge base. An

initial set of relation instances, built from Wikipedia InfoBoxes, is provided as an initial knowledge base. The relation extraction in KBP relies on a corpus of 1.7 million articles, which is much larger than earlier evaluations as MUC or ACE.

More precisely, the goal of the *Slot Filling* task is to collect attributes for entities of type *Person, Organization, Geo-Political entity*. Examples of relational attributes of each named entity type are presented in Figure 2.3. In total, 42 kinds of attributes are defined for these three entity types (KBP 2009).

Person: alternate_names, date_of_birth, age, place_of_birth, origin, date_of_death, place_of_death, cause_of_death, residences, ...
Organization: alternate_names, members, member_of, subsidiaries, parents, founded_by, founded, dissolved, headquarters, shareholders, website, ...
Geo-Political Entity: alternate_names, capital, subsidiary_orgs, top_employees, political_parties, established, population, currency

Figure 2.3: Attributes for named entities in KBP

Compared to ACE, in which relations between named entity pairs need to be explicitly expressed in the same sentence, KBP permits answers in different sentences, even promoting systems to search answers in the entire corpus by emphasizing on cross-document resolution other than information extraction from individual documents (Ji and Grishman, 2011; Min and Grishman, 2012).

2.1.4 TREC Series

The TREC campaigns started in 1992 and are still on-going annually. The Entity Track since 2009 is related to our interest of information extraction as well (Voorhees and Buckland, 2009, 2010, 2011). An entity is defined as a person, product, or organization with a homepage [1]. In the *Related Entity Finding* (REF) task, given an input entity, a target type and a narrative (nature of the relation in free text), a ranked list of homepages of related entities should be returned. A query example and its topic definition is given in Figure 2.4. The input entity is *Boeing-747*, and the target type is *Organization* of *airlines*. Therefore, a list of homepages of airlines which use *Boeing-747 planes* should be returned.

In total, 20 topics are considered in 2009, and 50 more are added in 2010. Topics contain Organizations that award Nobel prizes, CDs released by the King's Singers, and the like (Balog et al., 2010, 2011, 2012).

[1] A homepage is devoted to and in control of the entity and Wikipedia pages are accepted in 2009 but not any more since 2010

Query:
airlines that currently use Boeing-747 planes?

TREC topic definition:
<narrative> *airlines that currently use Boeing-747 planes* </narrative>
<entity_name> *Boeing 747* </entity_name>
<entity_URL> *clueweb09-en0005-75-02292* </entity_URL>
<target_entity> *organization* </target_entity >

Figure 2.4: One topic example from TREC 2009, Entity track

2.1.5 Evaluations in Specific Domains

Some evaluation tasks concentrate on specific domains, especially on biomedical domain. The BioCreAtIvE[2] challenge evaluation has been held since 2004 for evaluating IE systems applied to biological domain, detecting biological significant names such as gene and protein names and associating them to existing database entries and functional facts (e.g. protein-function). The i2b2 challenge [3] includes tasks of extracting medical concepts from patient reports, assigning assertion types for medical problem concepts and relation types between medical problems, tests and treatments (Uzuner et al., 2011).

Tasks in specific domain differ from those in open domain, since the information types and relations are specialized in this very domain. Proposed approaches include the use of supervised classifiers (Uzuner et al., 2010), hybrid systems combining rule-based methods and machine learning methods (Minard, 2012), or unsupervised models (Ciaramita et al., 2005).

2.1.6 A Brief Summary

In this thesis, we focus on the IE tasks in open domain. Therefore, the comparison of tasks and approaches presented here is only based on open domain IE. As a summary of the evaluation tasks presented above, templates are used in the beginning (early MUCs) for the extraction of a given event type. Then information is represented as relations, with a growth of the number of relation types, one objective being to have a rich variety of relations. In ACE, the task is supervised in the sense that the relation types are predefined and annotated in the corpus, whereas in KBP, a weak supervision is demanded, since once an ensemble

[2]Critical Assessment of Information Extraction systems in Biology `http://www.biocreative.org`

[3]Informatics for Integrating Biology and the Bedside `http://www.i2b2.org`

of attributes is defined for entity types, no extra annotation is required for the corpus. A list of related tasks and corresponding information representation is shown in Table 2.1.

Evaluation	Related Task	Information Variety
MUC (1987 - 1997)	Template Filling	tens of templates
	Template Relation	3 basic relation types
ACE (1999 - 2008)	RDC	5 categories, 24 relation types (See Figure 2.2)
TAC KBP (2009 - present)	Slot Filling	42 attributes for 3 entity types (See Figure 2.3)
TREC (1992 - present)	REF	70 topics (See Figure 2.4)

Table 2.1: Evolution of information extraction evaluations

Generally, we would like to divide information extraction tasks into three categories: supervised IE, semi-supervised IE, and unsupervised IE. The first two kinds of tasks concern the detection of limited types of relations, while the last kind of task is open for discovering new relation types. A brief illustration of these three different tasks is given in Figure 2.5.

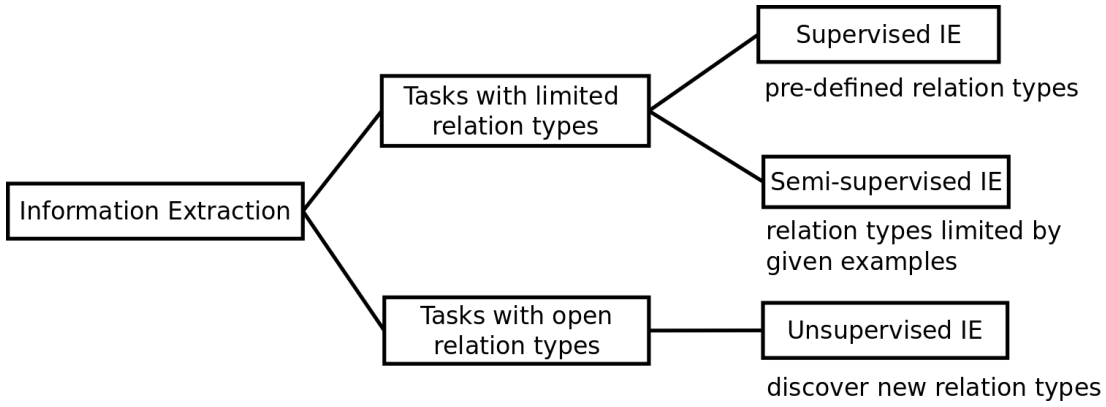


Figure 2.5: Three categories of Information Extraction tasks

More precisely, supervised IE refers to tasks such as RDC of ACE or *Template Relation* of MUC. Researcher often use manual engineering systems or supervised machine learning methods for detecting predefined types of relations in a corpus annotated with entities. Such systems will be discussed in section 2.2. Alternately, relation types themselves are not directly predefined in semi-supervised IE tasks but are often limited by an ensemble of candidates of relation instances, from which relations are learned. An example of semi-supervised IE could be the KBP *Slot Filling* task, which concerns learning fixed types of relations from an existing knowledge base, containing only relation instances without

annotated texts. This paradigm is often called *distant learning*. Other systems start from a seed of relation examples and then use bootstrapping methods to obtain more examples. Approaches linked to semi-supervised IE will be presented in section 2.3.

For supervised IE, manual engineered systems often require experts to construct precise templates, while supervised machine learning models need to be created from annotated corpora. On the other hand, for semi-supervised IE, relation types are always limited by the choice of seeds or resources. To go beyond the limits of these two kinds of systems, unsupervised IE developed in recent years is characterized by its ability of discovering a diversity of relations in open domain with minimal human supervision. It is to be noted that supervised machine learning methods can be as well applied in unsupervised IE tasks, to decide the validity of a relation for example. The task itself remains unsupervised as far as the types of relations are not fixed *a priori*. This thesis concentrates mainly on the design of unsupervised IE system. Existing methods about how information is extracted with current systems will be detailed in Section 2.4. The other issue occurring with unsupervised IE is how to organize similar information discovered in corpus of open domain. Therefore, techniques for organizing extracted information will be discussed subsequently in the end of Section 2.4.

2.2 Supervised Information Extraction

For supervised IE, the early systems proposed during MUC-7 in 1998 are mostly manually engineered systems (Aone et al., 1998; Brady et al., 1998; Huyck, 1998; Patten et al., 1998; Yangarber and Grishman, 1998) and include only few machine learning based systems (Miller et al., 1998).

IE² (Information Extraction Engine) is an example of manual engineering system developed by *SRA International, Inc.* (Aone et al., 1998). As first steps, **IE²** annotates named entities with a software that is capable of sub-typing and linking name aliases. Then its IE engine tags custom names including *AIR, GROUND, WATER, etc*, and also complex noun phrases and local links such as *employee_of, location_of, product_of, and owner_of*. It also includes a discourse module for co-reference resolution. At last, with all this information, templates are generated by mapping attributes from input files with customized rules configuration.

SIFT (Statistics for Information From Text) by *BBN Technologies* uses probabilistic models (Miller et al., 1998). The probability of a relation instance is learned using training examples annotated with semantic and syntactic information. The semantic training is

based on a manually annotated corpus of 500,000 words from New York Times, while *Penn TreeBank* (Marcus et al., 1993) is chosen as syntactic training data. The trained models were then used to search the most likely semantic and syntactic interpretation for any new given sentence. Relations can be directly extracted from the syntactic and semantic structures.

For the *Template Relation* task, both **IE²** and **SIFT** reach a practically operational performance, with F-Measure scores of 75.63% and 71.23% respectively.

Besides the generative approach used in **SIFT**, learning approaches using discriminative models, mostly based on kernels, are also developed in the following years. Kernels can be based on shallow parse trees to learn syntactic patterns from a set of already extracted relations, as in (Zelenko et al., 2003) for relations as *person-affiliation* and *organization-location*. The work of (Zelenko et al., 2003) was then extended by Culotta and Sorensen (2004) using a richer sentence representation and a feature-weighting framework based on dependency trees. Bunescu and Mooney (2005) obtained better extraction performance by concentrating on the shortest path between entities in the dependency graph. In addition to one-level kernels in previous work, Zhao and Grishman (2005) built multi-level kernels including tokenization, sentence parsing and deep dependency analysis, which allowed to use information from one level to overcome errors from another level. Other learning models have also been tested for the supervised IE tasks, such as maximum entropy based models (Kambhatla, 2004) or graphic models (Culotta et al., 2006; Rosario and Hearst, 2004).

Many experiments of these researches are based on corpus from MUC or ACE, where relation types are well defined. Satisfying performances were achieved by these kinds of systems. However, it is very time-consuming to build rules manually (**IE²**) or to annotate training corpus (**SIFT**) for these predefined relation types. The same time-consuming work needs to be repeated if one tries to apply these systems to other relation types in a different domain.

2.3 Semi-supervised Information Extraction

Semi-supervised information extraction does not rely on annotated relation types while the relation types could be either fixed by the selected initial seed examples or by an existing ontology resource. Usually, associated approaches go beyond the constraint of massive manual corpus annotation for one's task. Relation instances can be retrieved in large amount by bootstrapping from a restricted set of relation examples as initial input. Alternatively, larger input allows to reduce the times of iterations, sometimes with even one single

iteration for an extraction system. Large open resources such as WordNet, or Wikipedia are often used, either for learning patterns or for training statistical models. These two categories of semi-supervised IE approaches will be discussed in details hereafter.

2.3.1 Bootstrapping from Selected Seeds: Snowball Effect

As an early semi-supervised IE system, DIPRE (Brin, 1998) was designed for the purpose of extracting from the World Wide Web a relation characterizing books in the form of *(author, title)* pairs. More precisely, relation pairs are represented by a five-tuple pattern:

$$<\textit{order, urlprefix, prefix, middle, suffix}>$$

where *order* is a boolean value that defines the order of *author* and *title* in a pair, *urlprefix* matches the URL of the web document in which this relation instance is found, and the others define the context around the pair. The system starts with a small set of *(author, title)* pairs, and then searches on the Web all occurrences of these books, from which patterns of the relation are generated. These patterns are used to find new books and new occurrences. New relation patterns can be generated again. The procedure is repeatedly carried out until enough relation pairs are retrieved.

The underlying observation for this bootstrapping is the duality that a reliable set of patterns could be built from a good set of tuples and that a trustworthy set of tuples could be built by matching a good set of patterns. Therefore, reliable pattern generation algorithm and instance selection are important for the quality of bootstrapping. DIPRE generates patterns by grouping occurrences of relation tuples which share the same strings between instance pairs (the *middle* part in the pattern).

Yi and Sundaresan (1999) chose the same bootstrapping idea for extracting *(acronym, expansion)* relations. Besides the advantage of duality between related pairs and patterns, they make use of the duality between related pairs and the acronym *formulation rules*, which is a list of rules about how an acronym is formed from a given word. They also make more complex patterns by adopting HTML-Tagged patterns in addition to Text-Only patterns, since in Web documents, many *(acronym, expansion)* pairs are embedded directly as HTML tags and attributes as:

$$<a\ \textit{name="CSS" href="..."} > \textit{Cascading Style Sheet} $$

Double duality and rich patterns are proved to be useful for improving both the quality and the quantity of retrieved pairs.

Riloff and Jones (1999) introduced a multi-level bootstrapping approach based on their system $AutoSlog$ (Riloff, 1993, 1996a,b), which is used to extract relation patterns and semantic lexicons from untagged text. They add a ranking and selection heuristic to the multi-level bootstrapping procedure. After having patterns generated by $AutoSlog$, extraction patterns are scored in each iteration according to how many lexicons a pattern extracts, and only the best patterns are chosen for the next step. In the second level of bootstrapping (named meta-bootstrapping), the relations are scored based on the intuition that a noun phrase extracted by more patterns is more pertinent to a category than one extracted by only one pattern. Again, only best noun phrases are selected for the next iteration. This produces a snowball effect to enhance the pattern-relation duality in the sense that better noun phrases will be chosen iteratively by better scoring patterns.

Going further with the snowball effect, Agichtein and Gravano (2000) built the system *Snowball* by emphasizing the importance of pattern generation, which should be both flexible to capture most relation tuples hidden in text and in the same time selective to avoid invalid ones. Patterns are defined as a 5-tuple:

<left, tag1, middle, tag2, right>

where tag1 and tag2 are named-entity tags and the others characterize the context around. The contexts are weighted vectors, each component of which indicates the importance of the associated term. Several more complex confidence functions were also tested to evaluate patterns and relation tuples. Their experiments were based on *Organization-Location* relations.

Similar systems have been proposed using different information in patterns. Yangarber et al. (2000) adopted subject-verb like predicate arguments, supported by name normalization and syntactic analysis; Sudo et al. (2003) used subtree model based on arbitrary subtrees of dependency trees; Surdeanu et al. (2006) chose syntactico-semantic patterns, acquired by cooperation with text categorization techniques. Various ranking methods were also discussed in the above researches.

A common divergence problem which is known as the *Semantic Drift* disturbs such bootstrapping methods since general patterns have more tendency to be high scored after several iterations. In the system *NELL*[4], Carlson et al. (2010) came up with a coupled semi-supervised learning to alleviate this problem. In addition to a handful of seed examples, they use also an ontology defining target categories and relations, and a set of constraints to couple them, so that only patterns in agreement with all constraints can be chosen. For

[4]Never-Ending Language Learning project at Carnegie Mellon University `http://rtw.ml.cmu.edu/rtw`

example, the category of *Person* and *Sport* are mutually exclusive, similar to the function of named entity in earlier systems. Their experiments showed that the precision of the extraction can be significantly improved with coupled learning.

2.3.2 Learning from Large Open Resources: Distant Learning

An ensemble of relation seeds can be easily given by a user to target information types. However, the quantity and the diversity of relations are always limited by this manual supervision. Nevertheless, several sorts of open resources may contain abundant entity pairs where various relation types can be found. Machine learning methods can make use of on these resources as initial knowledge bases, the procedure of which is generally named *distant learning* or *distant supervision*. Such resources used include *WordNet* (Miller, 1995), *Wikipedia* (Wikipedia, 2004), *Freebase* (Bollacker et al., 2008), *DBpedia* (Bizer et al., 2009), *YAGO* (Suchanek et al., 2007), *OpenCYC* (Lenat, 1995), etc. These resources are constructed manually or semi-automatically with various objectives, not necessarily to annotate relations between entities. On the contrary, most of the entity pairs included in these knowledge bases are independent of relation instances in sentences. The assumption of *distant supervision* is that:

> "if two entities participate in a relation, any sentence that contains those two entities might express that relation" (Mintz et al., 2009)

Therefore, relation patterns can be learned from these automatically acquired sentences, in which relation instances are supposed to occur. In the next part of this section, some of these resources, adopted in current researches, are presented in more details.

WordNet is a lexical database of English, grouping nouns, verbs, adjectives, and adverbs into sets of cognitive synonyms (synsets), and providing interlink between these synsets to form semantic and lexical taxonomy structures. Snow et al. (2005) use *WordNet* as a learning source to automatically learn hypernym relations (*is a*). They use dependency paths to formalize the lexico-syntactic patterns between *is a* pairs. All sentences with identified hypernym pairs are then collected to train a classifier based on these patterns. The classifier succeeded to distribute from all patterns the high-scoring ones such as:

NP_Y, like NP_X:	N:PCOMP-N:PREP,like,like,PREP:MOD:N
NP_Y, called NP_X:	N:DESC:V,call,call,V:VREL:N
NP_X, is a NP_Y:	N:S:VBE,be,be,-VBE:PRED:N
NP_X, a NP_Y :	N:APPO:N

Wikipedia is a free multilingual Internet encyclopedia, collaboratively edited by volunteers from all over the world. Apart from its online text for searching and browsing, related information is accessible in many other ways, including links, taxonomic structures, InfoBoxes, etc. An *InfoBox* contains a concise table of the subjects' attributes, such as *map_size*, *population_of_California*. Wu and Weld (2007) started from the *InfoBoxes* to select schemata with relevant attributes to generate a data set for training. Then they train a classifier to learn the category of Wikipedia articles and another classifier to check if an sentence contains a given attribute. At last, one extractor for each attribute of each article category is learned to automatically create and complete *InfoBoxes*, say *population* of other cities or regions which are not in current InfoBox tables.

The *Wikipedia InfoBoxes* also served as a Knowledge Base to learn patterns for *Slot filling* task of KBP evaluation (see Section 2.1). For each slot, Grishman and Min (2010) chose one or two patterns from an *InfoBox* as initial seeds. Then they choose to start the bootstrapping steps from seed patterns rather than relation pairs so that more pairs of relations can be included for learning. The total number of patterns augments from 34 in the beginning to 970 after 3 iterations. Alternatively, Jean-Louis et al. (2011) took the whole *InfoBox* for patterns learning and made a single iteration for extraction. There is always a balance to be found between the number of initial seeds and the number of iterations. Generally, a large starting set of seeds allows to train a classifier in one pass, while in same time it limits the capacity of iteration because it is more time-consuming.

Freebase provides free online database of structured semantic data. Mintz et al. (2009) started from relations instances *(relation, entity1, entity2)* in *Freebase*, and then retrieve sentences in Wikipedia articles containing these relation instances, which are considered as positive relation examples for the training of extraction models training. Syntactical, lexical and named entity tag features are combined for the classifier learning. Riedel et al. (2010) also use *Freebase* as Knowledge base and then applied their approach to extract relations from newspapers corpus. Considering the fact that occurrences of entity pairs may appear in sentences which are not referring to the target relation in knowledge base, especially when the test corpus is not directly related with this knowledge base, they employed a more relaxed *expressed-at-least-once* assumption that:

> "If two entities participate in a relation, at least one sentence that mentions these two entities might express that relation."

An undirected graphical model was designed to predict in the same time the relations between entities and the sentences expressing these relations. They achieved more precise results with the relaxed assumption compared with the basic *distant supervision* assumption.

Table 2.2 shows a synthetic comparison of different techniques of bootstrapping methods and distant learning methods.

Bootstrapping (patterns)	strings	Brin 1998; Riloff and Jones 1999; Yi and Sundaresan 1999
	named entity	Agichtein and Gravano 2000; Carlson et al. 2010
	syntactic/semantic	Yangarber et al. 2000; Surdeanu et al. 2006
	subtree model	Sudo et al. 2003
Distant learning (resources)	WordNet	Snow et al. 2005
	Wikipedia	Grishman and Min 2010; Jean-Louis et al. 2011
	Freebase	Mintz et al. 2009; Riedel et al. 2010

Table 2.2: A variety of approaches for semi-supervised IE

2.4 Unsupervised Information Extraction

New approaches for Information Extraction have been explored during these last years, which aim at finding in texts relations between target entities or types of entities without any *a priori* knowledge concerning the type of the extracted relations.

Work in this area can be considered according to three main viewpoints. The first one regards the unsupervised extraction of relations as a means for learning general knowledge. This view has been developed both for learning "general world knowledge" through the concept of *Open Information Extraction* (Banko et al., 2007) applied for large-scale knowledge acquisition from the Web in (Banko and Etzioni, 2007) and in more restricted domains, as the biomedical domain, where such relation extraction is used for adding new types of relations between entities in an already existing ontology (Ciaramita et al., 2005). The second viewpoint tries to make it possible for users to specify their information needs in a more open and flexible way. For example, the *On-demand information extraction* approach (Sekine, 2006) aims at inducing a kind of *template* from a set of documents that are typically retrieved by a search engine from queries and these queries are specified by users as their target topics or relation types. Finally, the last viewpoint, less represented than the two others, considers unsupervised information extraction as a source of improvement for

supervised information extraction, so that coverage of models learned from an annotated corpus can be extended (Banko and Etzioni, 2008; Gonzàlez and Turmo, 2009).

These approaches are called Unsupervised Information Extraction, since the target relation types are not limited by human annotated examples (Supervised IE) or initial seeds (Semi-supervised IE). Even with different viewpoints or prospective applications, the main objective of unsupervised IE is to build a system which requires minimal human labor and which can discover various unforeseen relations in corpus. These relation types can be very diverse especially in open domain. Among different researches, relations are mostly represented in a binary way as a relation triple since one of the first unsupervised IE researches (Hasegawa et al., 2004), such as :

<Google, *be offer to buy*, Youtube>

This triple is characterized by first of all a pair of arguments, which could be named entities or noun phrases. These two arguments are linked together by a potential binary relation. With this definition, the task of unsupervised IE is to retrieve from a given corpus the largest number of valid relation triples, which can include all kinds of relations and argument pairs.

This simple but effective representation in a triple form is adopted by many researchers, exploring different categories of approaches. To discover unknown relations, clustering methods are first used to group similar relation tuples together, so that valid relations can be extracted by selecting the high-scored ones in clusters, such as the most frequent ones (Hasegawa et al., 2004). Methods based on clustering methods for relation discovery will be presented and compared in Section 2.4.1. In parallel, another way to extract relations is to start by a user-given query to focus on the topic of resulting documents, which gives an interactive orientation to unsupervised information extraction, such as in the systems of KNOWITALL (Etzioni et al., 2005) and *On-Demand Information Extraction* (Sekine, 2006). Query-based methods are presented in Section 2.4.2. More recently, researchers seek for a more scalable way of relation triple extraction, typically started with *Open Information Extraction* (Banko et al., 2007). The objective is to train a classifier to decide the existence of a valid relation in a triple and then this classifier is used as an extractor to retrieve from a corpus all relation triples considered as valid ones. Extractors are trained independently of relation types so that different kinds of relations can be found. Researches of this kind are detailed in Section 2.4.3. Other approaches, including generative models (Rink and Harabagiu, 2011; Yao et al., 2011) or rule-based systems (Akbik and Broß, 2009; Gamallo et al., 2012), are presented in Section 2.4.4.

Latest researches have also extended this binary representation to N-ary relation (Akbik and Löser, 2012), or to automatically create adapted templates (Chambers and Jurafsky,

2011) (Section 2.4.5). No matter which kind of representation chosen, the task of unsupervised IE is to make the implicit structure in the text more explicit, which is then useful either for constructing a general knowledge base or for being served as a supplement to supervised IE, or any other potential application.

2.4.1 Binary Relation Discovery with Clustering Algorithms

Clustering methods are adopted to group similar binary relations together, so that those relations instances which are grouped as relatively large clusters can be distinguished from other instances, and then are considered to be interesting relations. As one of the pioneer researches of this kind, Hasegawa et al. (2004) proposed three basic intuitional assumptions on the feasibility of clustering methods for unsupervised IE:

- pairs of entities occurring in similar contexts can be clustered;
- each entity pair in a cluster is an instance of the same relation;
- meaningful relations are frequently mentioned in large corpora.

Hasegawa et al. (2004) first tag named entities in a corpus with the extended named entity tagger OAK[5] (Sekine, 2001; Sekine et al., 2002). Then occurrences of named entity pairs within the same sentences are extracted. For the same types of named entity pairs, their similarities are calculated with the cosine similarity measure using a feature vector represented by a bag-of-word of named entities and all intervening words from all co-occurrences of these two named entities. Hierarchical clustering is then applied to group similar relation tuples together. At last, those formed clusters which have a size larger than a threshold are kept by the system, to pick useful and frequent relation tuples, as stated in the assumption. This threshold of a basic selection for valid relation instances is set at 30 in practice. The frequency of common words is counted for all combinations of named entity pairs in each cluster, and the most common words are considered as the characterization of this particular relation.

It should be noted that clustering methods have a dual role here, since once the relation instances are extracted, they are in the same time organized according to their similarities. This is not always the case for other kinds of relation extraction methods. In this section,

[5]OAK contains 150 types of named entities and can detect more specific entities such as *Military* and *Government*, which could be both *Organization* for a simple tagger. It can also link different values of the same entity together, as *George Bush* and *G. Bush*.

we concentrate on approaches about the extraction aspect, and more about relation organization issues using clustering techniques will be discussed later in Section 2.4.6 and then detailed in Chapter 4.

The variety of linguistic expressions around entity pairs makes it possible to discover potentially different mentions of relations in large corpus of open domains. For instance, different expressions of relations between organizations is obtained as the following:

<ORG>A</ORG> *be offer to buy* <ORG>B</ORG>
<ORG>A</ORG> *'s propose acquisition of* <ORG>B</ORG>
<ORG>A</ORG> *'s interest in* <ORG>B</ORG>

Even though with a successful discovery of non-predefined types of relations, a relatively low precision is reported by Hasegawa et al. (2004). Following researches have been investigated in several directions of advancement:

- How can representative patterns be defined, rather than simply bag-of-word?
- How to rank candidates for selection, instead of setting a basic cluster size threshold?
- How to group relations with an adapted clustering methods?
- How to give an appropriate label to each relation cluster?

Relation pattern choice Instead of using merely bag-of-words as in (Hasegawa et al., 2004; Chen et al., 2005), different kinds of patterns are adopted to enrich the representation, including lexical or syntactical types of information. Lexical patterns are used as a generalization of bag-of-words in (Rozenfeld and Feldman, 2006a) (URIES system). The context-describing patterns are learned and extracted from sentences, which forms the basis of the feature space of relation representation vector. Patterns are defined to be arbitrary sequence of elements in a sentence, such as *tokens, skips, slot marks* (e.g. *tokens* for character string or part-of-speech tag matching, *skips* describing gaps between tokens, and *slot marks* indicating relation argument position in the pattern). It should be mentioned that these patterns are not only used for describing the properties of relation candidates in the clustering stage, but also used for bootstrapping novel instances of identified relations, which is not feasible in (Hasegawa et al., 2004), due to the relatively low precision.

As an example of syntactical information in patterns, Shinyama and Sekine (2006) consider the text syntactically connected to an entity as *basic patterns* for this entity, such as *"Entity is hit"*, *"Entity 's residents"*, etc. If both entities of an entity pair share the same *basic patterns* with entities of another entity pair, they contain probably similar relations.

A statistical parser and a rule-based tree normalizer are used to obtain for each sentence a GLARF structure, which is a structure that normalizes several linguistic phenomena so that syntactic variety can be handled in a uniform way (Meyers et al., 2001). With GLARF structure for each sentence in documents, *basic patterns* are then generated for entities.

A combination of lexical and syntactic patterns are adopted by Bollegala et al. (2010), collecting lexical-syntactic information around two entities, for example *Entity1 acquisition of Entity2* as lexical pattern with words and *Entity1 NN IN Entity2* as syntactic pattern with part-of-speech information.

Ranking algorithm The ranking issue concerns the ranking of relation instances and in the same time the ranking of patterns. Hasegawa et al. (2004) set a threshold of frequency to filter non-frequent relation instances. Using a frequency-based measure, Rozenfeld and Feldman (2006a) give a high confidence to relation instances with entities not frequent by themselves in the corpus but relatively frequent as a pair. Others use frequency or frequency-based measures for pattern selection. Shinyama and Sekine (2006) give more weight to frequent entities in each event cluster for *basic pattern* generation. Bollegala et al. (2010) use frequency as a threshold for the selection of both entity pairs and patterns.

Another category of ranking measure is based on entropy. Chen et al. (2005) represent words as features, assuming that a feature is irrelevant if its presence obscures the separability of data set. The entropy of data set is calculated repeatedly by removing one feature each time, and is then compared with the total entropy. The features whose removal results in minimum entropy are considered as the least important ones. A similar entropy-based ranking measure is adopted in (Rozenfeld and Feldman, 2007).

Grouping methods Various grouping methods appear in existing researches. Since the number of relation clusters is unknown in advance for unsupervised IE, the adapted clustering algorithm must be able to deal with this issue. Hierarchical agglomerative clustering (HAC) does not require a predefined number of clusters, so it is adopted in (Hasegawa et al., 2004; Rozenfeld and Feldman, 2006a). There are generally three ways of similarity calculation for HAC: complete linkage (Hasegawa et al., 2004; Rozenfeld and Feldman, 2006a), average linkage, single linkage, which calculate respectively the maximum, the average and the minimum of similarities between data points in two clusters.

On the contrary, Chen et al. (2005) apply a stability-based criterion to automatically estimate the number of clusters; then K-means clustering is used from this estimated number. Rozenfeld and Feldman (2007) make a comparison among different hierarchical agglomerative clustering algorithms and K-Means. For HAC methods, all three ways of linkages

criterion were tested and a superiority of single linkage hierarchical clustering is demonstrated by their experiments.

Two levels of clustering are performed by Shinyama and Sekine (2006). Articles reporting the same event are first clustered using a bag-of-words approach for article discrimination, so that enough instances are collected to enable their *basic pattern* generation for each entity in basic clusters. Then a second step of *meta-clustering* is to group the events that contain the same relation.

Bollegala et al. (2010) propose the *relational duality* that a semantic relation can be either defined by entity pairs (*extensional* view), or by properties of patterns (*intensional* view). Therefore, a sequential co-clustering algorithm is used to simultaneously group the subset of patterns and the subset of entity pairs. Patterns and entity pairs are represented in a matrix as columns and rows, and the co-clustering groups iteratively the best patterns for entity pair and the best entity pairs for a pattern.

Table 2.3 synthesizes different options used by researchers for the clustering procedure.

	Patterns	**Ranking**	**Clustering**
Hasegawa et al. 2004	bag-of-words	frequency	HAC
Chen et al. 2005	bag-of-words	entropy-based	K-means
Shinyama and Sekine 2006	syntactic	frequency-based	two-level
Rozenfeld and Feldman 2006a	lexical	frequency-based	HAC
Rozenfeld and Feldman 2007	lexical	entropy-based	HAC, K-means
Bollegala et al. 2010	lexical-syntactic	frequency	co-clustering

Table 2.3: Comparison of different options of binary relation clustering

Labeling methods Some researches also make efforts for giving a name to each cluster formed, as a label of the relation instances inside the cluster. Hasegawa et al. (2004) choose directly the most frequent common words in a cluster by counting the frequency of common words in all combinations of named entity pairs in this cluster. A measure based on frequency of words in all patterns of related context is defined by Rozenfeld and Feldman (2006a) to give scores to features, so that the highest can be regarded as the relation label.

Alternatively, discriminative methods are used for relation label identification. Chen et al. (2005) use *Discriminative Category Matching*[6] to score features by combining local information of feature distribution within a cluster and global information across clusters. Then, features of words with the highest scores are chosen as labels for the clusters. Bol-

[6]*Discriminative Category Matching* is used for document classification to give weights to features of words based on their distribution (Fung et al., 2002)

legala et al. (2010) model this labeling problem as a discriminative feature selection issue in multi-class classification. Each entity pair in a cluster is represented as a feature vector, logistic regression methods identify lexical patterns which discriminate one cluster to the other, and the highest non-zero weighted lexical pattern is selected as the cluster label.

2.4.2 Query-oriented Unsupervised Information Extraction

Clustering methods for relation discovery are based on the assumption that interesting relations of diverse types repeat frequently enough in large corpus whereas query-oriented systems start from a query to create a more homogeneous corpus, so that relations in retrieved documents are mostly related to the keyword or the topic by this query. Query-oriented methods provide an interactive interface between the users and the extraction procedure, and for extraction tasks on different topics, one only needs to change the query keywords then the same extraction procedure can be applied. This is similar to bootstrapping methods (discussed in Section 2.3) in the aspect of providing *interested relations*. Nevertheless, bootstrapping methods start from examples of a fixed relation type while here, the downloaded documents of one query give only a focus topic for the relations.

Inside the documents retrieved by a query, words do not have the same distribution compared to those in the whole document collection. The difference of distribution in the corpus can be used to discriminate relations in retrieved documents, as in *On-demand information extraction* system. Alternatively, an automatic pattern-learning procedure can be carried out on the retrieved documents, and then used to get more extractions. Such systems include also KNOWITALL, URES and IDEX, which are presented further in the rest of this section.

On-demand information extraction In the *On-demand information extraction*, Sekine (2006) observed the fact that relevant documents retrieved with query keywords contain information about the salient relations of this topic. The author extracted those sub-trees of dependency that are relatively more frequent in the retrieved documents than in the entire corpus. The top-ranking sub-trees containing named entities are considered as patterns that indicate relations of the topic. These patterns are then applied to the original corpus to construct a table for each relation. Moreover, patterns are grouped into semantic pattern sets using a paraphrase knowledge base, so that a larger table for each relation is built.

KNOWITALL systems KNOWITALL (Etzioni et al., 2004a,b, 2005) learns from retrieved documents the corresponding relation patterns. Generic extraction patterns are defined by rule templates. For example, rule templates for the "of" relation could be :

NP1 *is the* P *of* NP2
P *'s of* NP1 *is* NP2

Here, P is a user given predicate, which could be *CEO* or *capital*. These rules are associated with queries to search sentences automatically from Web pages (e.g. "CEO of Amazon" or "capital of France"). Resulting Web pages are used to learn relation patterns in an unsupervised way, in contrast with traditional supervised models trained using manually tagged examples on small corpus. To guarantee the correctness of each extraction, the authors made the hypothesis that the more a relation is repeatedly extracted from distinct sentences in a corpus, the more probable is the correctness of the relation. Thus the confidence probability is computed, using *pointwise mutual information* between words and sentences estimated from Web search engine hit counts.

Later systems further improve the confidence calculation with URNS model (Downey et al., 2005, 2010), which quantifies relation confidence with redundancy information in text. In this so-called "balls and urns" model, each extraction is modeled as a labeled ball in an urn. Each label, either a relation instance or an error, may appear on a different number of balls in the urn, and the probability of one extracted element with the target label in the urn can be estimated with Bayes Rule.

One limit of early implementations of KNOWITALL is the requirement of large number of search-engine queries and page downloads, which can be very time-consuming at scale. A more recent version named KNOWITNOW (Cafarella et al., 2005) deals with this issue by using their own specialized search engine, Bindings Engine, which returns bindings to search queries efficiently, such as a list of proper nouns likely to be city names for a query like *cities such as*.

URES Compared to the rather selective patterns defined in KNOWITALL, which makes high-precision extractions but ignore most sentences, URES (Rozenfeld and Feldman, 2006b; Feldman and Rozenfeld, 2006; Rozenfeld and Feldman, 2008) aims at improving the recall. URES uses a more expressive pattern definition with lexical information (as for URIES system presented above in Section 2.4.1). For pattern learning, positive sets are generated by sub-string search of known instances of predicates. Meanwhile, negative sets consist of known false predicates. In the extraction step, each extraction is scored by calculating the ratio of positive matches on negative ones.

IDEX IDEX system (Eichler et al., 2008) considers all sentences in the retrieved documents that contain at least two named entities as a potential relevant relation. Then these sentences are treated with the Standford parser (De Marneffe et al., 2006) to build skeletons of simplified dependency trees, which are used for relation extraction. They focus on verb relations, and collect named entity pairs where at least the subject or the object is an named entity. Relations are grouped together if their similarity exceeds a predefined threshold.

2.4.3 Open Relation Extractors

Clustering-based methods for unsupervised information extraction require preparing in advance relation candidates (such as named entity pairs), while open relation extractors aim at making a single-pass extractor to retrieve relation instances from any corpus or Web pages.

Additionally, open relation extractors differ from query-based methods in that they loosen the topic restriction of the query, which means relation extraction from different topics will be carried out in the same procedure, rather than separated by different queries. All types of relations, as long as they are valid ones, are considered by open relation extractors.

Systems with open relation extractors include first of all TEXTRUNNER (Banko et al., 2007) of University of Washington, which trains a classifier to determine the dependency between noun phrases. Later systems by researchers in University of Washington make improvements using Wikipedia (WOE, Wu and Weld (2010)), or syntactic-lexical constraints (REVERB, Fader et al. (2011)). Alternatively, STATSNOWBALL starts from given seeds, and learns the extractor in a bootstrapping way. All these systems are presented briefly below.

TEXTRUNNER Banko et al. (2007) introduced an *Open Information Extraction* system, TEXTRUNNER, the object of which is to discover unanticipated concepts and relations from large corpus as the Web. Each extraction from a sentence takes the form of a tuple:

$$t = (e_i,\ r_{i,j},\ e_j)$$

where e_i and e_j denote entities, and $r_{i,j}$ denotes relations between them. Then a parser is used to obtain their dependency graph representation for each sentence. The tuple is labeled as a positive example if certain syntactic structure is shared by e_i and e_j, while as negative one if not. A Naive Bayes classifier is then trained using positive and negative examples, and then can be integrated to the extractor to determine if the extracted tuple is trustworthy or untrustworthy.

In a later version (Banko and Etzioni, 2008), rather than using a Naive Bayes classifier to give a label to the whole relation instance, the relation extraction is modeled as a sequence labeling problem to give labels to words in each sentence which is involved with a valid relation. Heuristics are applied to the PennTreebank to capture dependencies via syntactic or semantic role labeling, so that a set of positive and negative examples can be identified. Then a linear Conditional Random Fields (CRF) model is trained to tag each sentence. Relatively high precision is reported (88.3%), and the recall is improved from 23.2% to 45.2%, compared to the Naive Bayes approach.

WOE Wikipedia-based Open Extractor (WOE) proposed by Wu and Weld (2010) matches sentences in Wikipedia articles corresponding to relations expressed as attribute values in Wikipedia infoboxes. Then these sentences are used to train an unlexicalized and relation-independent extractor over parser features to extract relation tuples just as TEXTRUNNER. Two kinds of extractors are learned depending on the features, one called WOE^{pos} using only shallow features as POS tags, another called WOE^{parse} taking into account dependency trees. WOE^{pos} trains classifier over the context between noun phrases to decide if the text indicate a semantic relation, while WOE^{parse} checks whether the shortest dependency path between two noun phrases contains a valid relation.

Compared to TEXTRUNNER, a better precision is achieved by WOE^{pos}, which contributes mainly to an augmentation of F-measure by 14% to 34%. WOE^{parse} earns an F-measure which is 72% to 94% greater than TEXTRUNNER. However, TEXTRUNNER runs 30 times faster than WOE^{pos} due to its shallow parsing.

REVERB According to the observation in (Fader et al., 2011; Etzioni et al., 2011), TEXTRUNNER and WOE both suffer often from two types of errors : *incoherent extractions* and *uninformative extractions*. The first type of errors refers to phrases with no meaningful interpretations, while the second type concerns relations which omit critical information. Examples are given in their articles such as:

- "They *recalled* that Nungesser *began* his career as a precinct leader."
  incoherent extractions: (they, recalled, Nungesser), (Nungesser, began, his career)
- "Faust made a deal with the devil."
  uninformative extractions: (Faust, make, a deal)
  correct extraction: (Faust, make a deal with, the devil)

To solve the *incoherent extractions* type of errors, REVERB adds syntactic constraints for multi-word relation phrases. These phrases must begin with a verb and end with a

preposition, and the word sequence must be contiguous. For *uninformative extractions* type of errors, *light verb constructions* [7] are adopted to include nouns in relation phrases (e.g. *make a deal with* instead of *make*). Moreover, over-specific relation phrases are eliminated by an additional lexical constraint.

In the extraction step, REVERB first locates the longest word sequence started by a verb, which satisfies all constraints, and then searches the proper arguments around this word sequence. An ensemble of extraction from the Web and Wikipedia are then manually annotated, so that a classifier can be trained to assign a confidence score to each extraction.

A later version called R2A2 (Etzioni et al., 2011), introduces also an argument identification procedure, ARGLEARNER, to help deciding the bounds of two arguments, which is important especially when arguments contain prepositional attachments or a list of noun phrases. Better precision and recall are reported for both REVERB and R2A2 compared to TEXTRUNNER.

STATSNOWBALL *Snowball* system (Agichtein and Gravano, 2000) is built for bootstrapping iteratively from corpus specific types of relations starting from user given seeds. The patterns generated by *Snowball* are mainly based on keywords matching. Taking a similar iterative procedure, STATSNOWBALL (Zhu et al., 2009) extends the keywords matching of specific patterns to a general pattern learning. The sequence of part-of-speech between entities is used for general patterns. Then Markov Logic Network (Richardson and Domingos, 2006) is used for pattern learning and selection.

STATSNOWBALL performs joint references for pattern learning by Markov Logic Network (MLN) as well, taking in account in the same time entity-level model, sentence-level model, and page-level model. Entity-level model uses Logistic regression with the strong assumption that the relation between two entities is independent from other entities and relation keywords; Sentence-level model treats one sentence as a whole to detect jointly whether a relation exists between two entities and whether the context indicates the relation type using CRF; Page-level model considers further joint information from other sentences in the Web page, with MLN modeling correlated data. Both Logistic regression and CRF are a special case of MLN model. The system demonstrates the improvement of performance with this joint reference.

[7] A *light verb construction* (LVC) is a multi-word expression which combines a verb with a noun phrase or preposition (Stevenson et al., 2004)

2.4.4 Generative Models and Rule-based System

Generative models (Yao et al., 2011; Rink and Harabagiu, 2011) and rule-based systems (Akbik and Broß, 2009) have also been developed recently for unsupervised Information Extraction. Generative models are similar to methods that use clustering algorithms (Section 2.4.1) in the sense that they both require a large enough corpus so that reliable relation instances can emerge from all the instances. On the other hand, rule-based systems are similar to open relation extractors (Section 2.4.3) since they both make the decision (a valid relation or not) by relying on the information in each single sentence, using either a machine learning classifier or a set of handcrafted rules. These two kinds of approaches are presented below.

Generative Models for Unsupervised Relation Extraction

Generative models, such as Latent Dirichlet Allocation (LDA) (Blei et al., 2003), which are widely used in topic models (i.e. to detect in an unsupervised way the set of topics appearing in a collection of documents), can be applied to relation extraction as well. In traditional LDA, as illustrated in Figure 2.6, given D documents, N words for each document and K topics (a predefined number of topics), each topic is characterized by a distribution of words β_k, and each document by a distribution of topics θ_d. With the only observation $W_{d,n}$, the goal is to approximate posterior distribution using techniques as Bayesian inference, Gibbs sampling, expectation propagation, etc.

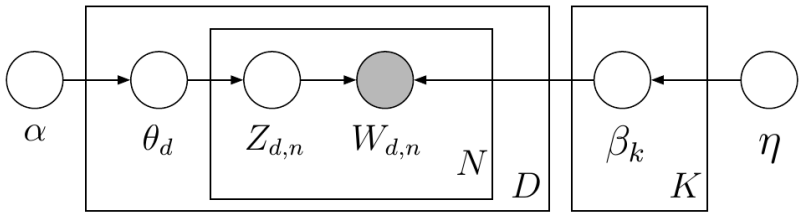


Figure 2.6: LDA model (Blei et al., 2003)

As a basic adaption of LDA to relation extraction, a topic represents a relation type between two arguments, and a distribution of this relation type over words in documents can be obtained. Rink and Harabagiu (2011) define a document with tokens in the entity pair and the context, and then LDA is processed on these *pseudo-documents*, so that clusters can be formed since relation arguments co-occur in *pseudo-documents*. Rink and Harabagiu (2011) use this as a baseline for their relation extraction evaluation. In their system of Relation Discovery Model (RDM), they consider the three parts in the relation tuple (two

arguments, and text around) separately as different distributions and jointly models these three distributions together.

Yao et al. (2011) represent each document as an ensemble of relation tuples, which include two named entities mentions and the dependency path between them. Instead of inferring distributions of words directly, they draw the distribution of features based on observed words, including POS, named entity, syntactic information, trigger words, etc. Another version of their system split features from the two named entities out of other features, which improve the recall in certain cases. A later improvement of their system (Yao et al., 2012) integrated as well more global features such as document themes and sentence themes, which increase both recall and precision.

Both Rink and Harabagiu (2011) and Yao et al. (2011, 2012) demonstrated that a relatively richer representation of LDA modeling can ameliorate system performance. In the same time, the richer the models are, the more complicated and costly the parameter estimations are.

Rule-based Relation Extraction System

Rule-based systems are well developed for supervised information extraction tasks, especially during MUC series. In recent years, researchers also developed systems of this kind for unsupervised information extraction tasks, constructing rules of grammatical structures which are independent of relation types.

WANDERLUST system, proposed by Akbik and Broß (2009), searches this kind of universally valid grammatical patterns using a deep linguistic analysis. For each sentence, formalized links are drawn for terms which are grammatically dependent, and if the path between two terms (*linkpath*) is valid, a semantic relation is considered to exist between two terms. A corpus of 10,000 sentences are manually annotated so that 46 valid *linkpaths* are generated, with which semantic relations can be extracted, such as those extracted by top valid *linkpaths* listed in the article for instance: *Is, IsCityIn, WasKilledBy, HadCaptured, FailedToDefeat, etc*. A rather high precision is achieved (82.1%), with a limited recall (16.1%), compared to a golden standard annotated by a human reader.

Generally, deep linguistic analysis is more time-consuming than shallow analysis. DepOE system (Gamallo et al., 2012) adopts a rather fast parser to guarantee system's scalability so that it can be applied on Web scale. With the parsed dependency trees information, verb clauses are discovered for each sentence, and for each clause, verb participants together with their functions are identified. At last, a set of rules is applied to clauses for relation triple extraction. Each extraction rule is a triple of tokens together with linguistic

information such as lemma, part-of-speech, etc. These rules are manually given for verb clauses, such as:

clause1: subject + verb phrase + direct object
clause2: subject + verb phrase + verb prepositional complement
clause3: subject + verb phrase + attribute

A better precision is obtained compared to REVERB system, due to its deep syntactic information.

2.4.5 Complex Relation Extraction

All the previously presented researches in this section concentrate on the extraction of binary relations. Some recent researches try to extract from documents information with different structures in the context of unsupervised Information Extraction. This work includes N-ary relation extraction (Akbik and Löser, 2012) and automatic template creation (Chambers and Jurafsky, 2011).

N-ary Relation Extraction

Binary relation extraction succeed at retrieving various true facts, however, these facts are sometimes incomplete, since more than two elements could be involved in a relation. The representation of N-ary relation or complex relation is adopted in (McDonald et al., 2005), with supervised methods trained on an annotated documents of biomedical domain. More recent work includes N-ary relation extraction in open domain, such as KRAKEN (Akbik and Löser, 2012).

To illustrate the limit of binary relation, the sentence below is given as an example in (Akbik and Löser, 2012) :

"Elvis moved to Memphis in 1948."

Binary relations extracted by two systems presented above REVERB (Fader et al., 2011) and WANDERLUST (Akbik and Broß, 2009) are the following:

REVERB : MovedTo(Elvis, Memphis)
WANDERLUST : MovedIn(Elvis, 1948)

Neither of these two relations are false, but both are incomplete. An ternary fact *MovedTo* involving three entities should be more accurate:

KRAKEN : MovedTo(Elvis, Memphis, 1948)

KRAKEN system takes as input a sentence which is processed by Stanford dependency parser. For words in a sentence, one word is directed either upward or downward to another word with a grammatical nature, if two words are linked via a typed dependency. Given sentence parsed, one fact phrase is identified as a chain of verbs, modifiers, prepositions linked by a certain typed dependency. Among words in a fact phrase, argument heads (e.g. *Elvis, Memphis* and *1948* for the example above) will be detected by following paths of dependency type (type-path), as subject, direct object, etc. A list of type-paths are defined for searching argument heads and all argument heads are returned for the fact phrase. Then the full arguments are detected by following downward links from argument heads and if at least one argument is found, it is extracted as a fact. Since the number of argument heads and arguments is not limited, any arity of relations can be extracted, all depending on the nature of the sentences.

Automatic Template Creation

Another way of enriching relation representation, instead of N-ary relations, is the use of templates. Traditional template-based methods in supervised information extraction tasks often require predefined templates for fixed types of relations. Chambers and Jurafsky (2011) propose a way of creating templates automatically for different types of relations from unlabeled documents.

For a specific type of event, a template includes a set of slots or semantic roles and the goal of the information extraction process is to fill these slots. Given a document, Chambers and Jurafsky (2011) first detect the event type and then apply related slots learned for this event type.

To do this, they start from clustering *event patterns* on the terrorism corpus of MUC-4, which contains event types as *bombing, kidnap, attack, arson, etc*. These *event patterns* are verbs, nouns in WordNet under Event synset, or verb and headword of its syntactic object. LDA and agglomerative clustering are tested to separate different *event patterns* into clusters and each cluster is an approximation of template topic. Then, more documents are retrieved for each cluster from a larger corpus by word matching, so that enough examples of *event patterns* are built to further learn semantic roles for each topic. A semantic role is a cluster of syntactic functions of the template's event words. Syntactical functions includes subject, object, preposition, and etc. For example, *bomb* is the subject of *explode* and *destroy*, and the object of *set_off*, or *target* is a preposition of *get_of*, and an object of *destroy*. These syntactic functions are clustered for each event cluster, so that well scored syntactic function clusters (e.g., *bomb*, *target*) are supposed to be interesting semantic roles

for this event type. In the extraction step, a document is matched with one event cluster first and these related semantic roles are applied for slot filling. Experiments have shown the system's capacity of constructing well-adapted templates, even for event types that do not exist in MUC-4 corpus.

2.4.6 Relation Organization

Since relation types are of such a diversity in open domain, it is a very important work to properly organize all extracted relations into categories in order to provide the end-users with a more concise and readable information. For example, relations containing similar expressions or synonymous phrases could be grouped together.

With regard to those approaches using clustering methods for binary relation discovery presented in Section 2.4.1 typically as the work of (Hasegawa et al., 2004), clustering methods play a dual role when introduced in unsupervised IE, since once the relation tuples are extracted, they are in the same time organized according to their similarities. For other kinds of systems, most of related previous work puts the emphasis on how interesting relation candidates can be discovered and extracted, while few researches provide a semantic organization. In TEXTRUNNER for instance, relations are only indexed for querying. In this section, we outline several researches on relation organization.

Results of Hasegawa et al. (2004) contain semantic variations among different instances in each cluster, since they consider each set of named entity co-occurrences as one relation at the start, and different contexts, including synonyms, may appear between the same named entity pair, such as *offer to buy* and *acquisition of*. Sekine (2006) created an off-line paraphrase knowledge base, using shared named entities to align the sentences from multiple newspapers reporting the same event. Relation patterns with the same named entity types, and keywords in the same set of paraphrase knowledge base linking two named entities, are places in the same pattern set. Eichler et al. (2008) used directly lexical information from WordNet to determine if two verbs are in the same synonym set.

Kok and Domingos (2008) proposed an approach, called Semantic Network Extractor (SNE), for building a network of semantic relations from the results of TEXTRUNNER. SNE aims at extracting high-level relations and concepts from the relation instances extracted by TEXTRUNNER through a co-clustering method based on Markov Logic that simultaneously generates classes of arguments and classes of relations. However, SNE does not exploit any lexical semantic resource, which prevents it from grouping some relations.

Min et al. (2012) also tried to cluster both relations and their arguments but they adopted a less integrated approach and heavily rely on lexical semantic resources built from corpora.

More precisely, they built a first set of relations, named Type A relations, with the objective of grouping relation instances that share the same linguistic form but have different arguments in order to define classes of equivalent arguments. A second phase of clustering exploits the classes of arguments of Type A relations for building Type B relations, which gather relations instances having similar expressions. The first clustering makes use of a large graph of similarity and hypernymy relations between entities built automatically by relying on distributional similarity and lexico-syntactic patterns while the resource used by the second clustering is a large base of paraphrases extracted from a corpus.

2.5 Overview of Our Unsupervised IE System

2.5.1 Positioning of The Thesis

This thesis is focusing on unsupervised information extraction tasks, where no *a priori* information is given. Therefore, the designed system should be capable of tackling all possible relation types, in contrast to supervised or semi-supervised systems (Section 2.2 and 2.3). Compared with unsupervised information extraction in biomedical domain (Ciaramita et al., 2005), we are more oriented to open domain, such as text corpus from newspapers or Internet, so that potentially involved topics could be very diverse, from economics to politics, sports, etc.

More precisely, we concentrate on the binary relations between entities represented as a relation triple (Hasegawa et al., 2004) with prospective applications such as technology watching. However, unlike (Hasegawa et al., 2004; Chen et al., 2005) which use clustering methods to group and discover interesting relation instances in the same time, we choose to separate the relation extraction procedure and relation clustering procedure. The relation extraction procedure is designed as a single-pass extractor to determine the validity of a relation instance such as TEXTRUNNER or REVERB. The separation of these two procedures makes it possible to design an extractor which is much more efficient and scalable. This thesis goes further than TEXTRUNNER and REVERB by additional researches on how the extracted relation instances can be organized in large scale according to their similarities. A brief synthesis of the positioning of this thesis is given in Table 2.4.

	Thesis positioning
Relation types	non predefined
Application domain	open domain
Relation representation	binary relation triple
Extraction method	a scalable extractor
Objective	relation extraction and organization

Table 2.4: The positioning of this thesis

2.5.2 Overview of The System

As stated above, the objective of this thesis is to extract and organize all possible types of relation instances in a large scale of corpus in open domain. The designed system contains two main parts, with each part divided into several steps, as illustrated in Figure 2.7.

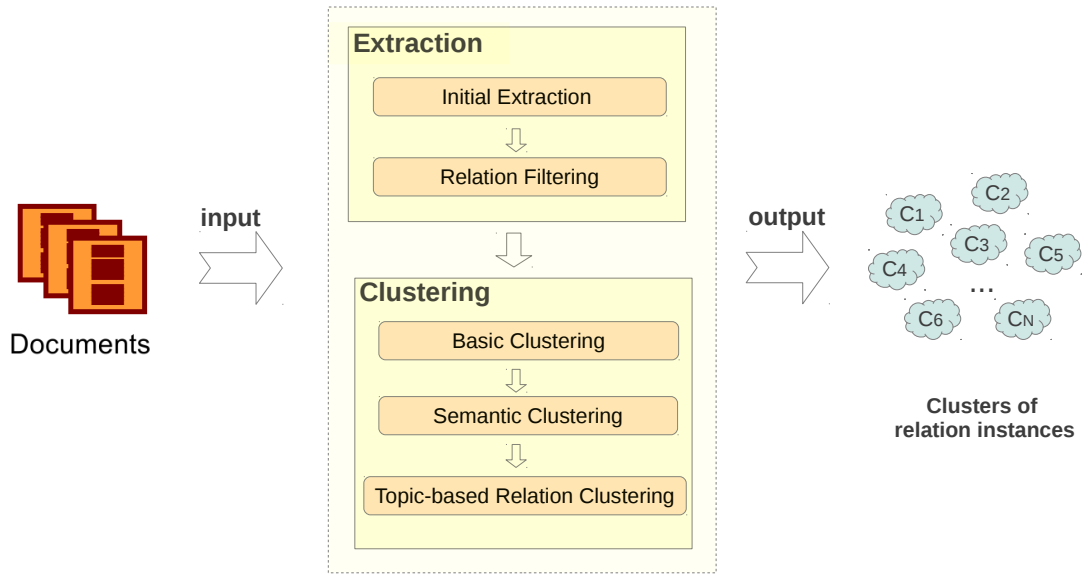


Figure 2.7: Overview of the system

The system starts from a large set of raw documents (*i.e.* unstructured documents). The objective is to generate a set of clusters of relation instances from the given corpus. The first part of the system, **Relation Extraction**, aims at extracting valid relation instances. To guarantee the scalability of the extraction procedure, we propose to perform it in two steps: a first step of *Initial Extraction* and a second step of *Relation Filtering*. *Initial Extraction* is executed with a defined relation prototype and minimal extraction criteria to extract all candidates of relation instances. It involves only very limited criteria so that it can be very efficient and can insure abundant relation instances and variety. More emphasis is then put

on the second step (*Relation Filtering*): the filtering of invalid relation instances from all candidates. Combination of heuristics and machine learning methods are experimented for filtering false relations. Heuristics are efficient for removing large amount of false relations while machine learning models make a more refined selection on valid relation instances.

The second part of the system, **Relation Clustering**, refers to the organization of extracted relation instances according to their similarities. Considering the large amount of extracted relation instances, we divide the relation clustering task into different steps as well. More precisely, a primary categorization is given to different relation instances according to the types of named entity pair. Inside each relation category, relation instances are first grouped by the similarities of their linguistic expressions (*Basic Clustering*) and then similarities of their semantic meanings (*Semantic Clustering*). At last, thematic information is also used to give a topic-based organization of relation instances which also help form more precise semantic clusters (*Topic-based Relation Clustering*).

The relation extraction part can be applied to any size of corpus, either a massive corpus with tens of thousands of documents or even one single article or paragraph, which means that the procedure is independent of the corpus size. However, the *Relation Clustering* part requires a certain number of extracted relation instances so that their similarities can be compared to form further relation clusters.

The final system output contains clusters of relation instances, with each relation instance considered as valid relation by our filtering procedure and each relation cluster supposed to represent one single semantic concept between two types of named entities. Relation extraction and relation clustering procedures are described in the next two chapters and evaluations and results are shown after that.

Chapter 3

Relation Extraction

Relations are represented in various ways in existing work of unsupervised information extraction. This thesis concentrates on the binary relation extraction between named entity pairs. A terminology defining in a precise way the different notions associated with relation extraction is first given in the beginning of this chapter. A prototype is defined to characterize relations and to initially extract candidates of relation instances with basic criteria. Then the focus will be on how to select valid relation instances among all initially extracted candidates. A filtering procedure combining heuristics and machine learning methods for eliminating invalid relation candidates is experimented and then applied on all candidates. Performance of this filtering procedure is also compared with existing systems.

Contents

3.1 Relation Definition 40

3.2 Terminology of Relation Extraction 41

3.3 Relation Characterization and Extraction 42

3.4 Filtering by Heuristics 47

3.5 Filtering by Machine Learning Models 49

3.6 Comparison with Other Systems 59

3.7 Application of Relation Filtering 62

3.8 Conclusions and Perspectives 64

3.1 Relation Definition

The various forms adopted for relation representations by different researches have been discussed in Chapter 2. In most of the unsupervised information extraction domains, tasks are often simplified to unsupervised binary relation extraction, which are more tractable, and in the same time are open to support various tasks, including general world knowledge construction (Banko et al., 2007), improvement of supervised relation extraction (Banko and Etzioni, 2008; Gonzàlez and Turmo, 2009), existing ontology mapping (Soderland et al., 2010), or applications of Question Answering systems. Nevertheless, it is essential to know or to define what a *relation* is before the start of any concrete work.

According to the Oxford dictionary [1], the term *relation* is defined as:

- the way in which two or more people or things are connected;
- a thing's effect on or relevance to another.

In tasks of unsupervised information extraction, an instance of binary relation is a triple which consists of two *arguments* connected by a *relation phrase* (Grishman, 2012), concentrating mostly on those relations expressed explicitly in the phrase. The arguments can be characterized by named entities (Hasegawa et al., 2004; Chen et al., 2005; Shinyama and Sekine, 2006), or, more generally, by noun phrases (Rozenfeld and Feldman, 2006a; Banko et al., 2007; Bollegala et al., 2010).

A **Relation** can be regarded as the way how two **Arguments** are connected together, this connection reflecting the effect of one argument to the other, the fact that one argument is an attribute of the other, or that one thing applicable to one argument is relevant to the other argument, etc. The definition allows to decide whether a valid relation exists between two given arguments with no constraints on the relation types. The relation types can be very diverse, such as the frequent "acquisition" relation between two companies, or the headquarter location of a company:

- <ORG>A</ORG> *be offer to buy* <ORG>B</ORG>(Hasegawa et al., 2004)
- <ORG>X</ORG> *headquarters in* <LOC>Y</LOC>(Bollegala et al., 2010)

Non-classical relation types between certain entities are concerned as well, such as the relation between a person and an empire:

[1] `http://oxforddictionaries.com`

- <PER>Napoleon</PER> *dissolved* <ORG>Holy Roman Empire</ORG>(Banko, 2009)

The objective of this thesis is to extract all these *classical* and *non-classical* relation types so that the defined relation prototype should loosen enough to allow the considerations of unforeseen relation types. The formal relation prototype is presented in Section 3.3, along with the introduction of the whole processing pipeline. Because of the fact that different terms are adopted in existing work, the terminology we use in this thesis is first defined in Section 3.2. Then, filtering methods for relation validity identification will be presented in Section 3.4 and Section 3.5, respectively for filtering heuristics and machine learning methods. In the end of this chapter, relation filtering performance will be compared to existing systems in Section 3.6, and results of applying filtering procedure will be shown in Section 3.7.

3.2 Terminology of Relation Extraction

Different terms have been used for relation extraction in open domain while there are often some ambiguities if we apply terms of one research to another. In order to make all the following presentation more clear, we propose in this section a specification of the terminology used to characterize the different notions of a relation. The different notions are illustrated on a standard example of "acquisition" relation between companies:

"In 2002, IBM decided to buy PricewaterhouseCoopers Consulting for $3.5 billion."

The first task of this thesis is to extract **Relation Instances** for different types of **Semantic Relations**, which are defined as:

- **Semantic Relation**: or the **Relation Type**, which is the semantic concept of the relation between two entities (e.g. *buy*)
- **Relation Instance**: an instance of a **Semantic Relation** expressed explicitly in a sentence, containing three elements:
  - **Arguments**: two named entities (e.g. *IBM*, *PricewaterhouseCoopers Consulting*)
  - **Mention**: the phrase which characterize the relation type around two arguments, mainly in the middle part between two arguments (e.g. *decided to buy*)

  - **Context**: context of this relation instance, which contains the content words of the neighbouring sentences (e.g. *..., IBM, decide, buy, PricewaterhouseCoopers, consulting, ...*)

Once relation instances are extracted, they are represented as **Instance Triples**:

- **Instance Triple**: a triple in the format of **Label(E1, E2)** used to represent a relation instance, where **Label** is a tag for the relation type, **E1** and **E2** are two arguments (e.g. *buy(IBM, PricewaterhouseCoopers Consulting)*)

In unsupervised information extraction tasks, the relation types are not pre-defined, so that the corresponding labels can not be known in advance. It is either given during human annotation of relation instances or generated in the clustering step. In addition, the same instance triple can be used for different instances sharing the same label and the same arguments.

Relation instances belong to different categories according to the named entity types of arguments:

- **Relation Category**: the pair of named entity types that categorizes the relation types (e.g. ORGANIZATION–ORGANIZATION)

Therefore, **Relation Extraction**[2] is the extraction of **Relation Instances** from corpus and **Relation Clustering** is the process of grouping relation instances by their similarities to form clusters.

3.3 Relation Characterization and Extraction

3.3.1 Relation Prototype

As previously stated, the arguments of a binary relation could be named entities or noun phrases. The choice of noun phrases provides a wider consideration to cover more candidate triples, while named entity allows a better separation of different relation types. This thesis takes place in a large context with the global objective of developing an unsupervised information extraction process for addressing technology watch issues. The process is based on the extraction of relation instances by the co-occurrence of two target arguments in sentences. Consequently, named entity is chosen for argument selection to guarantee a

[2]NB: Technically, it should be named as **Relation Instance Extraction**. We keep the term **Relation Extraction** since it is already widely used in the literature.

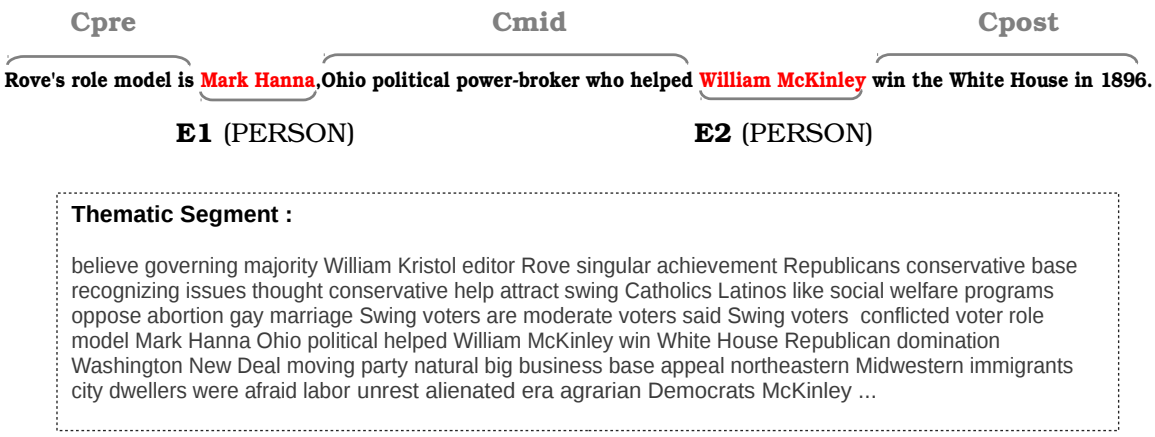


Figure 3.1: Example of extracted relation

more meaningful pair of target arguments. Afterwards, the basic intuitional extraction idea is to focus firstly on simple cases to counterbalance the difficulties raised by the heterogeneous nature of the global approach. "Simple cases" means here relations that occur within a sentence whose arguments are rather easy to identify and instances whose linguistic expression is concise enough to be easily delimited and to avoid coreference phenomena concerning their arguments.

More formally, relation instances extracted from texts are characterized by three elements: the arguments, the mention and the context (See Figure 3.1).

- **arguments**: a pair of named entities (E1 and E2);
- **mention**: the linguistic form of the relation instance. It refers more precisely to the way the relation is expressed in the local sentence. As relation extraction is based on the presence of two named entities in a sentence, the mention of the relation instance is made of three parts of this sentence:
  - *Cpre*: part before the first entity (E1);
  - *Cmid*: part between the two entities;
  - *Cpost*: part after the second entity (E2).

  The core expression of the relation is generally conveyed by *Cmid*, while *Cpre* and *Cpost* are more likely to bring local context elements that are used for detecting its similarity with other relations in the perspective of their clustering.

- **context**: thematic segments where the sentence of the relation instance is found. Each document is separated into multiple thematic segments, which refer to the context in which one relation instance occurs. Each thematic segment contains a list of content words which characterize the theme of this segment, and this word list offers a more global information to the extracted relation instance.

It should be noted that such relation takes a semi-structured form, since one part of its definition is characterized with elements coming from an already existing ontology (named entities) while its other part only appears under a linguistic form.

3.3.2 Initial Extraction of Relation Candidates

In this thesis, we are interested in the information extraction task in open domain and at a large scale. Therefore, we chose, for all our experiments, a subpart of the AQUAINT-2 corpus, which contains an ensemble of 18-month news articles from *New York Times*[3]. We concentrate our interest on relation types involving named entity types such as *persons* (PER), *organizations* (ORG) and *locations* (LOC).

As we stated in Section 2.5.2, we start our extraction tasks by a procedure of *Initial Extraction* in order to obtain a large number of candidates of relation instances. Before the *Initial Extraction* of relation candidates, a procedure of linguistic preprocessing is applied to the documents in order to obtain the linguistic information for representing the elements of relation instances.

Linguistic preprocessing More precisely, this procedure includes named entity recognition for the target types of entities, part-of-speech tagging and lemmatization for normalizing the three parts of the linguistic description of relations. This process is performed using the OpenNLP tools[4]. The linguistic preprocessing concerns also the thematic segmentation to associate an ensemble of thematic words with each relation instance as its context, for which the segmenter tool LCseg[5] is adopted. Since thematic information is not used for relation filtering but for relation clustering, details about thematic segmentation will be presented in the next chapter.

Initial Extraction The step of initial extraction of relation instance candidates involves very limited constraints: all pairs of named entities with the target types are extracted if

[3]This corpus of 18-month newspaper of the *New York Times* contains 159,400 documents in total.
[4]`http://opennlp.sourceforge.net`
[5]`http://www1.cs.columbia.edu/nlp/tools.cgi`

the two entities appear in the same sentence with at least one verb between them. In the case of more than two named entities in one sentences, all possibilities of named entity pairs are treated independently with this constraint. Thus, a large amount of candidates are extracted, with the volumes illustrated in Table 3.1.

Relation category	**Number of candidates**
LOC – LOC	116,092
LOC – ORG	57,092
LOC – PER	78,845
ORG – LOC	71,858
ORG – ORG	77,025
ORG – PER	73,895
PER – LOC	152,514
PER – ORG	126,281
PER – PER	175,802
ALL	929,404

Table 3.1: Volume of extracted candidates of relation instances

3.3.3 Error Analysis of Relation Instances

Candidates of relation instances are extracted in very large quantities as shown in Table 3.1. Since the restrictions applied for candidate extraction are rather limited, it is important to verify whether a *true* relation exists or not between each named entity pair in candidate sentences. Embarek and Ferret (2008) show that 79% of extracted candidates using this heuristic are *true* relations in the biomedical domain. However, this does not guarantee the same performance for open domain because of the heterogeneous topics of corpus, so that the error rates of these candidates should be analyzed.

After a first inspection of relation candidate samples using a web-based annotation tool, we observed that a significant number of candidates do not contain a true relation between their entities. Therefore, we decided to characterize these errors and then to evaluate the quantity and the influence of each kind of errors. According to our observations, false relations appear frequently in the following three situations:

- Discourses : It concerns particularly those relation categories with *person* as the first named entity type in the pair, and another entity is mentioned in the discourse of this *person* but is not connected by an explicitly expressed relation;

  - e.g. The Bruins are recruiting several Notre Dame players, and Knights coach *Kevin Rooney* said this was the first time *Dorrell* has been on campus.
  - e.g. "I vote for President Mubarak because I could not find any candidate more handsome than *Hosni Mubarak*", said *Mohany Ziad* , 48 , as he cast his ballot in the Cairo neighbourhood of Torah and then pressed his neighbours to vote the same way.

- Long Distances : We observed that instances of true relations generally have the two arguments of the named entity pair relatively near to each other in the sentence, whereas if there is a long distance between them, there is little chance that these entities are involved in a valid relation or at least, it is not obvious to determine this validity;
  - e.g. *Pentagon* officials applauded the capture of al-Hassan, who was listed as number 36 on the list of 55 most-wanted Iraqis that the American government compiled after the fall of *Baghdad* in April 2003.

- Over-complex linguistic forms: True relations are often expressed with rather simple linguistic forms. When too many verbs appear between the named entity pair, the chance of a true relation becomes low. Moreover, multiple verbs make the semantic meaning of the relation confusing.
  - e.g. "Guards Troy Hudson and *Fred Hoiberg* combined for 25 points after scoring only three in *Minnesota*."

These three types of errors above can be detected by their characteristics, such as discourse words, distances, etc, so that they can be removed efficiently in large quantity by configured heuristics. However, other observed errors are more delicately linked to the mentions of relation instances, such as the following examples:

- e.g. "Two divers from the *National Marine Fisheries Service* were returning to the dock near *Jenner* when Gieseke made his way back."
- e.g. " "I heard a huge explosion," said Yigal Vakni , an Israeli at the *Hilton* who spoke to *Israeli Army Radio*."
- e.g. "*Snipers* stood ready; crowds in *Iraq* can become targets."
- e.g. "Since *Meyer* took over, *Florida* hasn't had a player involved in an altercation."

- e.g. "After Jeff Bagwell popped out, *Beltran* stole second and *Lance Berkman* was walked intentionally to bring Kent to the plate."

Relation instances of these kinds of errors can be eliminated with more refined filtering techniques. Rule-based models search to define patterns such as *subj-vp-dobj, subj-vp-vprep*, etc (Gamallo et al., 2012). Systems as TEXTRUNNER try to learn the dependency information among words (Banko and Etzioni, 2008). We propose to train machine learning models which are based on only shallow linguistic information to automatically learn patterns as features for the determination of valid connection between two given named entities. Therefore, a two-step filtering procedure combining heuristics and machine learning models is designed as shown in Figure 3.2. These two steps of filtering will be detailed in the following two sections.

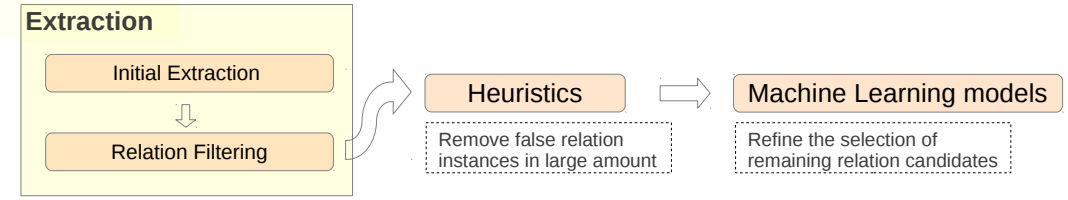


Figure 3.2: Two-step filtering for initially extracted relation candidates

3.4 Filtering by Heuristics

Referring to the three categories of errors with large quantities of instances, heuristic with empirical discriminative criteria is an effective and efficient way for removing them. Consequently, three heuristics were defined to filter out corresponding relation candidates which are in the above three situations:

- **Discourse** : Eliminate relation instances that contain a discourse related verb between the two entities (*say*, *present*);
- **Distance** : Limit the maximum distance between the two entities to 10 words[6], since effective relations become very rare beyond this empirical limit;

[6]The threshold value 10 is given empirically. Hasegawa et al. (2004) sets a maximum of 5 words but counts only context words while Yao et al. (2011) requires that the dependency path between two named entities must be shorter than 10.

- **Verb** : Allow only one verb between the two entities to avoid a too complex syntactic structure between them, excluding modal verbs and auxiliary verbs as *be*, *have* and *do*.

These three heuristics were applied first on a sample of 8,000 relation candidates for statistical analysis of each relation category. Table 3.2 details the effects of each heuristic by giving the filtering ratio of relation candidates.

Relation category	Filtered / Kept	Discourse	Distance	Verb
LOC – LOC	4287 (54%) / 3713 (46%)	440	3548	2763
LOC – ORG	4097 (51%) / 3903 (49%)	488	3224	2650
LOC – PER	4790 (60%) / 3210 (40%)	1636	3352	2638
ORG – LOC	4225 (53%) / 3775 (47%)	643	3324	2869
ORG – ORG	4169 (52%) / 3831 (48%)	627	3123	2810
ORG – PER	4541 (57%) / 3459 (43%)	1541	3155	2859
PER – LOC	4209 (53%) / 3791 (47%)	905	3199	2813
PER – ORG	3888 (49%) / 4112 (51%)	952	2742	2566
PER – PER	4444 (56%) / 3556 (44%)	1290	3109	2741

Table 3.2: Effects of the application of filtering heuristics on a sample of 8,000 relation candidates for each relation category

It is obvious to see from Table 3.2 that the application of these three heuristics globally reduced the volume of relation candidates by about 50%, compared to the initial extraction. For each relation category, the second column shows the number and the proportion of relation candidates filtered and kept using all the heuristics together. The remaining three columns give the number of filtered relation candidates by each individual heuristic, considering that one candidate may be covered by more than one heuristic. The distance limit and the only-one-verb limit have obviously an important filtering effect while the discourse heuristic offers a complementary impact for short and simple phrases between named entity pairs.

In addition to quantitative information about the reduction of the volumes of relation candidates, it is important to identify the quality of these filtering heuristics. In order to evaluate it, a subset of 50 randomly selected relation candidates of each relation category were manually annotated to verify their validity. The results of this annotation are detailed in Table 3.3.

Table 3.3 details true and false relation candidate proportions for both relation instances filtered and kept by heuristics. The first thing to observe is that, among those relation candidates filtered by heuristics, a very high percentage of them are indeed false ones (74% for category PER – LOC and 98% for category LOC – LOC).

Relation category	Filtered		Kept	
	true	**false**	**true**	**false**
LOC – LOC	1 (2%)	49 (98%)	9 (18%)	41 (82%)
LOC – ORG	4 (8%)	46 (92%)	8 (16%)	42 (84%)
LOC – PER	3 (6%)	47 (94%)	2 (4%)	48 (96%)
ORG – LOC	7 (14%)	43 (86%)	14 (28%)	36 (72%)
ORG – ORG	6 (12%)	44 (88%)	20 (40%)	30 (60%)
ORG – PER	4 (18%)	46 (92%)	20 (40%)	30 (60%)
PER – LOC	13 (26%)	37 (74%)	40 (80%)	10 (20%)
PER – ORG	12 (24%)	38 (76%)	40 (80%)	10 (20%)
PER – PER	5 (10%)	45 (90%)	14 (28%)	36 (72%)

Table 3.3: Evaluation of filtering heuristics for each relation category

What can be also seen is that the relation categories with *location* as first named entity type have very low chance to contain a true relation. This phenomenon can be explained by the fact that, in a true relation, the first entity should have an agent role in the sentence whereas location names often occur in adverbial phrases when they are at the beginning of sentences. These cases could be detected using a deeper syntactic analysis but such analysis is relatively costly for the amount of data in open domain, and this thesis concentrates on information extraction methods without deep linguistic analysis. Considering this observation, relations between entities involving a *location* as first entity are excluded from the following steps.

For the remaining 6 relation categories, the chance of being true relation augments by 2 to 3 times for relation candidates kept by heuristics compared to those filtered by heuristics. Nevertheless, these evaluation results also show that the ratios of false relations after filtering remain important, varying from 20% (PER – LOC and PER – ORG) to 72% (PER – PER and ORG – LOC). Therefore, further treatments are necessary to separate more precisely true relation instances from false ones.

3.5 Filtering by Machine Learning Models

The results in Table 3.3 demonstrate the utility of filtering heuristics. However, they are limited to three categories of errors (Section 3.3.3) and the results indicate that these heuristics are not sufficient to guarantee a high proportion of correct relations (*i.e.* high enough for the following steps of the unsupervised information extraction process). To solve other types of errors, an additional filtering method using statistical machine learning models is applied to refine the relation candidate selection. To do this, a training corpus was first an-

notated manually; then different statistical models were experimented with different ways of representing true and false relations.

3.5.1 Relation Annotation

A web-based annotation interface is first built for the annotation of relation instances, as shown in Figure 3.3.

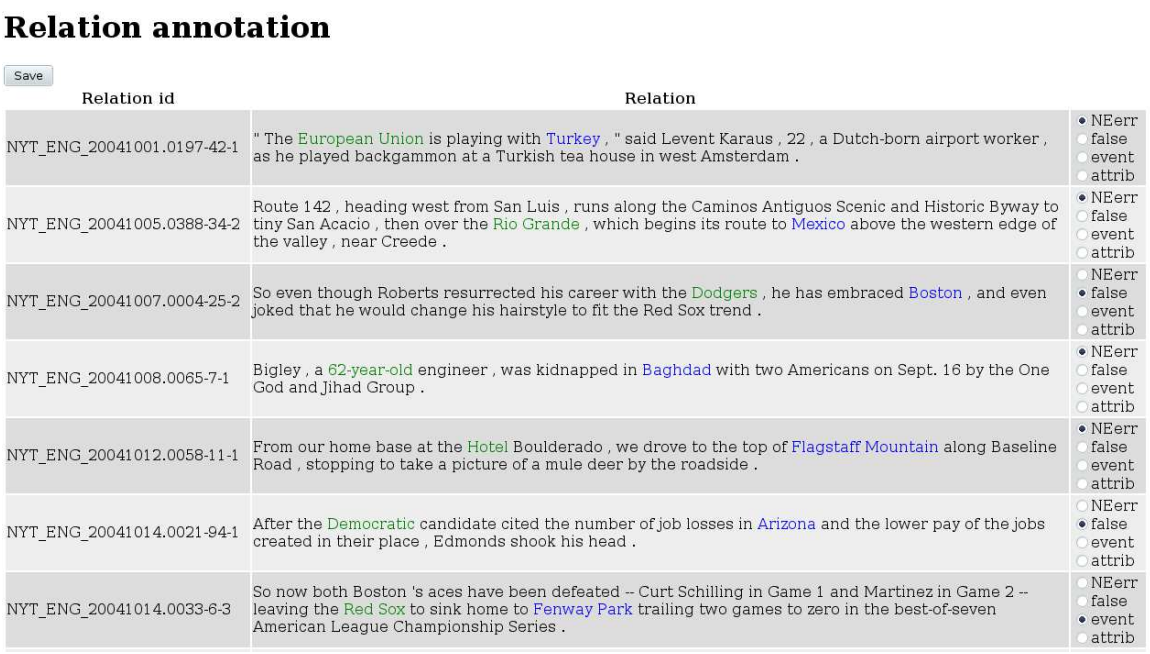


Figure 3.3: Interface for the annotation of relation instances

There are three columns in this interface. The first column contains ids of relation candidates. The whole sentence of the relation candidate is then shown in the second column. Considering that one sentence may include more than two named entities, an additional colour is applied on the active named entity pair, with red colour for *persons*, green colour for *organizations* and blue colour for *locations*. The last column is for giving a manual tag to the associated relation candidate. In total, four kind of annotations are distinguished:

- NEerr : one or both named entities in the pair are incorrectly recognized;
- False : no true relation exists between two correctly recognized named entities;
- Event : one named entity is connected to the other by an event-based relation;

- Attrib : one named entity is an attribute of the other.

In this stage of the thesis, we tried to investigate two types of true relations: an event-related relation is a dynamic event that connects two arguments while an attribute-related relation is a static attribute of one argument characterizing the other, such as the following three examples:

- **Event**: "*Bank of America* which **acquired** *Fleet Bank* for 48 billion last April."
- **Event**: "*George Bush* delivered his victory **speech** in *Washington D.C.* on Nov 3 2004."
- **Attribute**: "*Ianthe Brautigan* whose **father** is *Richard Brautigan* and who wrote you that ..."

The first two examples concern the event-related relations, which imply either a *direct event* or an *indirect event*. A *direct event* is the kind of events which happen directly between two arguments (e.g. *acquire(Bank of America, Fleet Bank)*) while an *indirect event* is in the situation where the event carried out by one argument is linked to another argument (e.g. *deliver_speech(George Bush, Washington D.C.)*). The last example is an attribute-related relation (i.e. *father_is(Ianthe Brautigan, Richard Brautigan)*)

The distinction between event-related and attribute-related relation is made because of the fact that, in an applicative context of competitive intelligence, event-related relations are generally of greater interest for end-users than attribute-related ones. Annotation results show the distribution of these two types of relations. However, our current model is not designed to distinguish these two types so that both types are merely considered as true relations.

Corpora for training and testing were then built manually using this interface. It is important to note that the removal of large amount of false relation instances by filtering using heuristics makes this annotation work easier to be realized. In addition, the remaining relation instances contain a more balanced number of true and false relation instances. In total, 200 relation instances for each target relation categories are randomly selected from the relation instances after filtering heuristics for annotation. Results of this annotation are presented in Table 3.4.

The figures in Table 3.4 show that about 20% relation candidates contain the named entity recognition errors. These relation instances were removed from the corpus to avoid introducing too much noise in the training data. More globally, it can be observed that about half of these annotations involve true relations, with a significantly larger number

Relation category	NE errors	False	Event	Attribute
ORG – LOC	35 (18%)	88 (44%)	56 (28%)	21 (11%)
ORG – ORG	28 (14%)	94 (47%)	70 (35%)	8 (4%)
ORG – PER	36 (18%)	92 (46%)	46 (23%)	26 (13%)
PER – LOC	62 (31%)	36 (18%)	88 (44%)	14 (7%)
PER – ORG	36 (18%)	44 (22%)	95 (48%)	25 (12%)
PER – PER	39 (20%)	79 (40%)	77 (38%)	5 (2%)
All	236 (20%)	433 (36%)	432(36%)	99 (8%)

Table 3.4: Manual annotation of 200 relation candidates for each category

of event-related relations compared to attribute-related relations: 432 instances versus 99 instances. The filtering classifier does not aim at distinguishing event-related and attribute-related relations so that they are both used as positive examples for machine learning.

Finally, Table 3.4 also shows that the volumes of annotated relation instances is not big enough for each relation category to train a classifier for each independently. Nevertheless, most of extracted relation instances rely on an implicit hypothesis assuming that the first entity had a verbal agent role in the sentence. Therefore, all these six target relation categories should have similar linguistic features for statistical models, so that one classifier can be trained for all the relation categories together.

A further examination of the annotated corpus led to find 5 duplicate relation instances for the whole 1200 annotations, including one annotated as *NEerr* for relation category ORG – LOC, two annotated as *false* for relation category ORG – ORG and two annotated as *true* (*Attrib*) for relation category ORG – PER. Duplicate relation instances were as well removed to avoid training bias.

Consequently, the remaining corpus for machine learning was composed of 960 relation candidates, as detailed in Table 3.5. This final annotated corpus contains 529 true relation instances and 431 false ones, which is a corpus with well-balanced positive and negative examples as data sets for learning a classifier.

3.5.2 Binary Classification Models for Relation Candidates

The objective of the classifier we develop here is to determine whether two named entities are connected by a true relation. This can be modeled as a binary classification problem for each individual relation candidate using features of these named entities and other information around. Interesting features are selected by their relevance to the existence of an effective relation but not the relevance to relation types. Therefore, the words themselves are not chosen as features to avoid an over strong connection of the classifier to the semantic

Relation category	False	True
ORG – LOC	88	77
ORG – ORG	92	78
ORG – PER	92	70
PER – LOC	36	102
PER – ORG	44	120
PER – PER	79	82
All	431 (44.9%)	529 (55.1%)

Table 3.5: Corpus finally used for classifier training

meanings of relation instances. Features concentrate on part-of-speech sequences around the two arguments to learn useful patterns of POS sequences and potential perturbation such as punctuation marks are adopted as well to learn the influence of noises.

Different machine learning models were experimented including Naive Bayes, Maximum Entropy (MaxEnt), Decision Tree model and Support Vector Machine (SVM). In order to compare these four different classifiers, the same set of features was adopted for training. Different ways of combining features were tested to find the most characterizing features for deciding the validity of relation and the one that led to the most stable performance was finally chosen. These finally used features are detailed below, illustrated by the following example sentence:

"A native New Yorker , Thomas Thacher graduated from Yale in 1938 and from its law school in 1942."

- type of named entities E1 and E2;
  (e.g. <E1, person>, <E2, organization>)

- Part-of-Speech (POS) of words between the two entities, using a binary feature for each pair $< P_i, POS_i >$, as well as bigrams of POS between the two entities, using a binary feature for each triplet $< P_i, POS_i, POS_{i+1} >$ (with P_i, the position of the current word in *Cmid*, $i \in [1, 10]$);
  (e.g. <1, VBD>, <2, IN>, <3, NULL>, <4, NULL>, ... <1, VBD, IN>, <2, IN, NULL>, <3, NULL, NULL>, <4, NULL, NULL>, ...)

- POS of the two words before E1 and the two words after E2, both as unigrams $< P_i, POS_i >$ and bigrams $< P_i, POS_i, POS_{i+1} >$ ($i \in \{-2, -1, 1, 2\}$) ;
  (e.g. <-2, NNP>, <-1, ",">, <+1, IN>, <+2, CD>, <-1, NNP, ",">, <+1, IN, CD>)

– POS sequence for words between E1 and E2 encoded as a binary feature;
(e.g. <ALL, VBD, IN>)

– number of tokens between E1 and E2;
(e.g. <nTokens, 2>)

– number of punctuation marks (comma, quotation mark, parenthesis ...) between E1 and E2.
(e.g. <nStops, 0>)

The longest distance among two arguments is 10 since the corpus was previously filtered using heuristics. Therefore, the size of POS sequence is normalized to 10. For sentences where the $Cmid$ part is shorter, special symbols "NULL" are used as features such as $< P_i, NULL >$ or $< P_i, NULL, NULL >$. The same symbol "NULL" is used if there are less than two words before E1 or after E2.

In these experiments, the Naive Bayes, Maximum Entropy, and Decision Tree were implemented using MALLET (McCallum, 2002), while the SVM model was implemented with SVM^{light} with the default configuration(Joachims, 1999).

Considering the relatively small size of the annotated corpus, the corpus was split into 10 parts so that a 10-fold cross-validation can be applied to evaluate these different statistical models. During each round, 9 parts were used for training and the last part for testing. This procedure was repeated 10 times so that each part was used for training and for testing at least once.

Standard measures of *Accuracy*, *Precision*, *Recall* and *F1-measure* was used and defined as:

$$\begin{aligned} Accuracy &= \frac{|N_{positive}| \cap |N_{true}| + |N_{negative}| \cap |N_{false}|}{|N_{true}| + |N_{false}|} \\ Precision &= \frac{|N_{positive}| \cap |N_{true}|}{|N_{positive}|} \\ Recall &= \frac{|N_{positive}| \cap |N_{true}|}{|N_{true}|} \\ F_1\text{-}measure &= \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \end{aligned} \tag{3.1}$$

where $N_{positive}$ and $N_{negative}$ are the set of relation instances found as true and false by the statistical models, while N_{true} and N_{false} are the set of references manually annotated true and false.

Table 3.6 gives the results of these evaluation measures, calculated as macro average values on the 10 rounds[7].

Model	Accuracy	Precision	Recall	F_1-measure
Naive Bayes	0.637	0.660	0.705	0.682
MaxEnt	0.650	0.665	0.735	0.698
Decision Tree	0.639	0.640	0.784	0.705
SVM	**0.732**	**0.740**	**0.798**	**0.767**

Table 3.6: Evaluation of statistical classifiers

These results clearly show that the SVM classifier obtains the best performance, a result that is generally obtained by similar work about relation extraction. The results obtained by Naive Bayes, MaxEnt and Decision Tree have similar F-measure performances, with a relatively higher recall by Decision Tree and better precision by Naive Bayes and MaxEnt. Additionally, it is worthy to note the balance between precision and recall for all types of classifiers.

3.5.3 Sequential Model for Machine Learning Filtering

Binary classification model trains a classifier to decide the relation validity for each relation candidate as a whole. Alternatively, the relation validity detection can be modeled as a sequence annotation problem by adopting the BIO encoding model (Ramshaw and Marcus, 1995).

More precisely, four types of labels are used for tagging words in relation candidates:

- O: words not related to a relation or named entity;
- NE: named entity that is the arguments of the potential relation (E1 or E2);
- B-REL: first word after E1 inside a relation;
- I-REL: continuation of a relation after B-REL.

Consequently, each relation candidate is annotated as a sequence of labels. Figure 3.4 and Figure 3.5 illustrate how sentences are annotated, in one case for a true relation and in the other for false relation.

In general, two categories of label sequences are used during the annotation:

[7]Macro average is computed by simply taking the average of the precision and recall of the classifier on different rounds

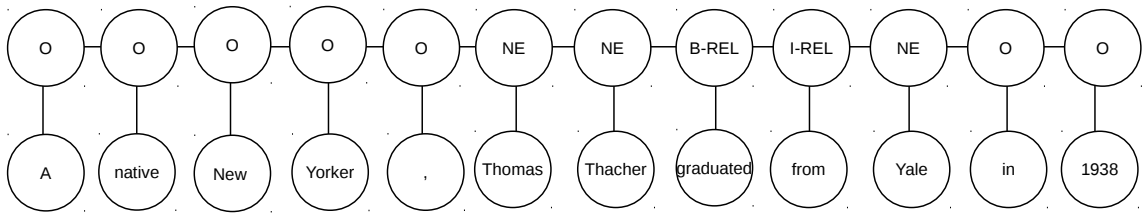


Figure 3.4: Sequential representation of sentence annotation for a true relation

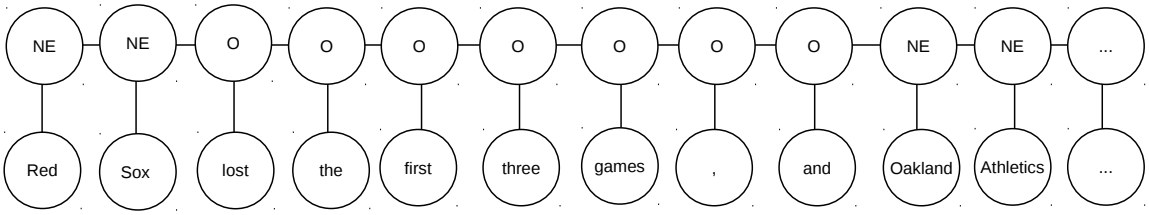


Figure 3.5: Sequential representation of sentence annotation for a false relation

- NE - B-REL - I-REL* - NE, if this relation instance implies a true relation
- NE - O* - NE, if this relation instance implies a false relation

Each category of label sequence contains a variable number of I-REL or O depending on the expression of the relation.

Even though there are only two categories of label sequence during annotation step, it is theoretically possible for the classifier to generate sequences with other combinations of these four label types. But in practice, a well-trained classifier does not produce configurations other than the two presented above: for instance (O – NE – B-REL – O – O – NE – O) is not possible since, in the training corpus, B-REL is always either followed by one NE or by at least one I-REL. This observation makes the decoding procedure (i.e. the identification of true or false relations by the label sequence obtained) much easier.

For sequential data labeling problems, Conditional Random Fields (Lafferty et al., 2001) (CRFs) provide a form of undirected graphical model that defines the distribution over a label sequence given an observation sequence[8] and can be trained on the annotated relation instances by giving selected features. Different feature sets were tested, especially those related to the number of neighbour words considered and their named entity types. The performance of different feature combinations are compared and the feature set which reaches the best and most stable performance was adopted. For each word in the sentence, features are generated as:

[8]CRFs outperform Hidden Markov Models (HMMs) by relaxing the dependence assumption which is required by HMMs for tractable inference. It also avoids the label bias problem of some directed graphical models such as Maximum Entropy Markov Models.

– POS of the current word, the previous one and the following one;

– bigrams of POS <POS_{i-1}, POS_i > , with i=-1,0,1 (0: current word; -1: previous word; 1: following word);

– entity type of the current word and the 6 previous and following words. This type is equal to NULL when the word is not a named entity.

This linear CRF was implemented with the Wapiti tool (Lavergne et al., 2010). Results were evaluated with the same cross-validation procedure as presented in Section 3.5.2. Table 3.7 gives the details of results and it shows that the CRF classifier slightly outperforms the SVM classifier in F-measure. Moreover, the same balance between precision and recall can be observed for sequential labeling model. The improvement of F-measure primarily comes from the amelioration of precision[9]. Precision has more importance than recall for unsupervised information extraction in open domain because of the huge amount of data processed. Therefore, this sequential model was chosen as the final statistical model for the second step of filtering.

Model	**Accuracy**	**Precision**	**Recall**	**F_1-measure**
SVM	0.732	0.740	0.798	0.767
CRF	0.745	0.762	0.782	0.771
Banko and Etzioni (2008)	/	0.883	0.452	0.598

Table 3.7: Evaluation of statistical classifiers

TEXTRUNNER (Banko and Etzioni, 2008) adopted a similar approach. The results they obtained is presented in the last line of Table 3.7. Our results, by comparison, show a better F-measure, with a much better recall and a slightly low precision (A more detailed comparison of the performance achieved by our systems and other systems will be presented in the next section.)

There are several differences between our relation extraction system and TEXTRUNNER. The first difference is that TEXTRUNNER identifies noun phrases as arguments while in our case, arguments concentrate only on named entities. Taking noun phrases as arguments makes it possible to consider more candidates while using named entities as arguments simplifies to some extent the determination of valid relation. Generally, named entities are more reliable than noun phrases as arguments so that the trained classifier will

[9]The difference of performance in terms of F-measure between the results obtained by CRF and SVM is very slight and should be tested for its statistical significance, even if our training set is not small. We chose here CRF rather than SVM by considering its better performance for the precision measure.

concentrate the mentions of relation instances other than considering the reliability of arguments (examples of unreliable noun phrases as arguments are shown in next section). This can partially explain why we obtain a much better recall.

Another difference of TEXTRUNNER is that it also tags the words that are not related to the relation as "*O*" even these words are inside an argument pair with a true relation, for example in the sentence:

"Tim*/ENT* Berners-Lee*/ENT* is*/O* created*/O* with*/O* having*/B-REL* invented*/B-REL* the*/ENT* WWW*/ENT*."(Banko, 2009)

The words "is", "created" and "with" are not related to the relation "invented" so that they are annotated as "O".

For sentences where there are more than two arguments, all arguments could be included in the relation tuple. Given the following sentence:

"Google*/ENT* announced*/O* that*/O* it*/O* acquired*/B-REL* YouTube*/B-NP* for*/B-REL* an*/ENT* astonishing*/ENT* $1.65 billion*/ENT*."(Banko, 2009)

The extracted tuple by TEXTRUNNER is *(Google, acquire (arg2) for, YouTube, $1.65 billion)*.

The objective of TEXTRUNNER is to tag all words in each sentence to identify arguments and related phrases for true relations whereas our strategy is to simplify the problem by treating all pairs of arguments separately. Hence, the only objective is to decide whether the sequence of words between two arguments does or does not yield a valid tag sequence, which indicates a true relation. To tag all words in sentence, the classifier trained in TEXTRUNNER is based on a corpus syntactically analyzed to indicate the dependencies among arguments while no syntactical information is used during our classifier training so that relation instances even with no syntactical dependency between two arguments are not excluded in our extraction.

Concerning the relation categories, it is also interesting to notice that we performed also experiments by removing the named entity type as a feature and replacing it by a generic "ENTITY" tag (e.g. <E1, ENTITY>, <E2, ENTITY>) to mark the presence of a named entity. The results are shown in the line of CRF_{ENT} in Table 3.8, compared with the original CRF classifier using named entity types as features (CRF). We can see that there is only a slight decrease of the *precision* compared to the classifier using named entity types as features while the performance of *recall* measure is even improved. This indicates a promising extensibility of our classifier to other categories with different named entity types.

Model	Accuracy	Precision	Recall	F_1-measure
CRF	0.745	0.762	0.782	0.771
CRF_{ENT}	0.740	0.752	0.789	0.768

Table 3.8: Comparison for statistical classifiers using named entity types as features or not

3.6 Comparison with Other Systems

The evaluation of filtering performance can be more illustrative when compared with existing systems, even though the comparison is not always direct because of the different relation prototypes defined in different researches. A series of systems are proposed by researchers in University of Washington, such as TEXTRUNNER (Banko and Etzioni, 2008), WOE (Wu and Weld, 2010), and REVERB (Fader et al., 2011). All their systems are capable to extract relation tuples of two arguments (noun phrases) which are linked with a meaningful relation explicitly expressed in a phrase. In the work of (Fader et al., 2011), the evaluation is based on a set of 500 sentences sampled from the Web using Yahoo's random link service[10]. All these systems and their variations were applied on this Yahoo sentence set and then the extraction results of all systems were pooled together for human annotation. The performance of each individual system was evaluated against the human annotated relation instances. Hence, the comparison here is done by applying the whole initial extraction and filtering procedure on these 500 sentences (*sentence set*) and then evaluating the results with the manually labelled reference (*annotation set*)[11].

In the *annotation set* of (Fader et al., 2011), there are in total 2,474 instances, 621 of which are annotated as true. Since named entities are used in our extraction procedure for locating relation tuples while all noun phrases are considered in their experiments, we first focus on relation instances that have named entities as arguments. The named entity module of OpenNLP tools was used to recognize named entities from the Yahoo sentence set. The results show that there are only 8 relation instances with named entity couples, which is not sufficient for a meaningful comparison. As a result, all noun phrase pairs in the *annotation set* are considered as pseudo named entity pairs; then among all sentences in *sentence set*, we relocated the sentence which each noun phrase pair belongs to; Finally, the whole extraction procedure was applied to check if a reliable relation between these two pseudo named entities is expressed in the relocated sentence. The statistical model CRF_{ENT}, which was trained without named entity types, was adopted in this case. Due to text format and noun phrase alignment issues, only 2412 noun phrase pairs were successfully located in

[10]It is available at `http://random.yahoo.com/bin/ryl`.

[11]Sentence set and annotation set are available at `http://reverb.cs.washington.edu`.

their corresponding sentences, among which there are 606 true relation instances. The statistics about the original *annotation set* and the part used for our extraction are given in Table 3.9.

Annotation Set	Total	True	False
Original	2,474	621	1,853
Evaluation	2,412	606	1,806

Table 3.9: Annotation of all extracted relation instances on the set of 500 sample sentences

REVERB is reported to achieve a precision of 0.8 for the best 30% extractions. In order to obtain the total extraction and its performance, the REVERB tool was re-applied to the Yahoo sentence set[12]. The total performances of REVERB and our extraction procedure are detailed in Table 3.10.

Systems	Positive	True Positive
REVERB	711	359
Our system	1,654/1,064	380

Precision	Recall	F_1-measure
0.505	0.578	0.539
0.357	0.627	0.455

Table 3.10: Comparison between REVERB and our system: evaluation by the reference from REVERB systems

REVERB succeeds in extracting 711 relation instances, with 359 of them being correct according to the reference. In the case of our filtering procedure, the first step, heuristics, removes 758 relation candidates including 95 true candidates. For the remaining 1654 candidates, 1064 of them were kept by our statistical filtering (these two numbers are both shown in the second column of Table 3.10). Although our filtering procedure has a small superiority on recall measure, REVERB obtains a better precision measure.

For a relation extraction system, there are two things to be guaranteed: first, the mention of a relation instance should indicate a true relation; second, the arguments should be valid roles that are linked by this relation. Comparing with systems using named entities as arguments, the choice of noun phrases as arguments is a double-edged sword since, on one side, it is capable of detecting more relation candidates while on the other side, it adds more complexity to the classifier to determine the valid arguments[13]. Figure 3.6 shows two examples that are annotated as false relations in their *labeled set* while our system chose to

[12] REVERB is available at `http://reverb.cs.washington.edu`. The adopted version is ReVerb 1.3, which was the latest version at the time of this experiment.

[13] REVERB added a procedure of *Argument Extraction* to learn the boundaries of corresponding valid arguments for a given relation mention (Etzioni et al., 2011).

keep them as positive relations. These two relation candidates are false relations because the related noun phrases are not valid arguments. However, our filtering system assumes that both arguments are named entities therefore both valid people or things. Therefore, our classifier concentrates more on the mentions of relation instances rather the arguments. Once the mention of a relation instance is considered as a valid expression, this relation is considered as a true one. This is one reason our filtering system has an inferior performance in precision measure than REVERB when applying our approach on their corpus.

"Among the other companies that got a boost from restructurings , **Sara Lee Corp.** *gained 17 percent for* **the week** after announcing it will sell some business and cut costs to raise $ 3 billion for a stock buyback ."

"Together , **the year 's two moves** *have raised prices 14 percent at* **the wholesale level** , and 7 percent at retail , according to Black ."

Figure 3.6: Negative relation instances in REVERB reference

We also applied REVERB on our reference of relations between named entity pairs to make a balanced comparison. From the 960 annotated relation instances (Section 3.5.1), there are 1790 relation instances extracted by REVERB. For each relation instance extracted by REVERB, we match it to the sentence which it belongs to and check if the arguments match our annotated named entities. Since the boundaries of the arguments are different between the named entities annotated in our reference and the noun phrases in extracted instances by REVERB, we consider that there is a match between the arguments from the two systems if the noun phrases have a non-null intersection with named entities. REVERB extracted 237 relation instances in which both arguments match annotated named entities in our reference, which means for 960 relation instances in our reference, REVERB detect 237 of them as true relations. Among these 237 relation instances, there are 197 correct answers according to our reference, which gives a high *precision* of 0.810. However, it can only discover 36.3% of true relations between named entities in our reference while our filtering procedure can detect 78.2% of them. This confirms the previous assumption, similar as the comparison with TEXTRUNNER, that the choice of named entities as arguments makes the classifier concentrate more on the mentions of relation instances for relation discovering which can then improve the *recall*. Table 3.11 shows the detailed results about this comparison.

System	Accuracy	Precision	Recall	F_1-measure
Reverb	0.953	0.810	0.363	0.501
Our system	0.745	0.762	0.782	0.771

Table 3.11: Comparison between REVERB and our system: evaluation by our annotated reference

3.7 Application of Relation Filtering

With the filtering heuristics and the statistical filtering discussed above, the whole extraction and filtering procedure was applied on AQUAINT-2 corpus in four steps:

1. initial extraction, only based on the the co-occurrence of two named entities with the target types and the presence of at least one verb in between;
2. application of three filtering heuristics for eliminating efficiently a large number of false relations with a good precision;
3. application of a machine learning filtering to distinguish more finely true relations from false ones;
4. elimination of duplicate relation instances.

The last step of duplicate elimination comes from the observation of the presence of a certain number of identical relation instances. These relation instances are often sentences from articles about the same subject, or from some regular journal remarks with very formatted expressions. Hence, the filtering procedure is completed with a final deduplication step for discarding these redundant instances. This deduplication step is implemented in an efficient way by identifying and grouping together relation instances whose similarities reach a maximal similarity threshold (1.0 in this case) and keeping only one representative element for each group. Markov Clustering is used for this grouping, which will be detailed in Chapter 4 about relation clustering. This deduplication operation is put at the end of the relation extraction process because this procedure is more costly than the other filtering operations. Hence, it is better to execute it on smaller corpus.

Table 3.12 shows detailed information about the volumes of processed relation instances for each step of filtering procedure, starting from all initially extracted relation candidates of Table 3.1. It can be noted that this whole filtering procedure puts aside a large number of the initially extracted relations, keeping only 24% of relation instances for our six relation categories. Finally, the remaining volume of relation instances is 165,708,

with at least ten thousand instances for each relation type, which is *a priori* sufficient for the experiments of our further work on relation clustering. The context of our work is the processing of large text collections characterized by informational redundancy, as on the Web for example. In this context, the ensemble of true relations removed by our filtering procedure is not an obstacle for our relation extraction system.

	Initial Extraction	**Heuristics**	**Classifier CRF**	**Deduplication**
ORG-LOC	71,858	33,505 (47%)	16,700 (23%)	15,226 (21%)
ORG-ORG	77,025	37,061 (48%)	17,025 (22%)	13,704 (18%)
ORG-PER	73,895	32,033 (43%)	12,098 (16%)	10,054 (14%)
PER-LOC	152,514	72,221 (47%)	55,174 (36%)	47,700 (31%)
PER-ORG	126,281	66,035 (52%)	50,487 (40%)	40,238 (32%)
PER-PER	175,802	78,530 (45%)	42,463 (24%)	38,786 (22%)
All	677,375	319,385 (47%)	193,947 (29%)	165,708 (24%)

Table 3.12: Relation volumes after each filtering step

Our two steps of filtering (filtering heuristics and statistical filtering) were evaluated separately with results shown respectively in Table 3.3 and Table 3.7. Since the statistical filtering was applied on all relation instances kept by filtering heuristics, it is also important to evaluate the global performance of our two steps of filtering, especially for two things: the overall recall after two steps of filtering and the ratio of true relation instances filtered in these two steps. From the existing filtering evaluation for each single filtering heuristic and statistical filtering, the global recall is estimated to be 0.559, with a precision[14] of 0.762. Among all discarded relation instances in the two filtering steps, the ratio of true relation instances is estimated to be at 24.0%. Results are shown in Table 3.13. The detailed calculation for the overall performance of the two steps of filtering is given in Appendix A.

	Precision	**Recall**	**F1-measure**	**TN ratio**
CRF	0.762	0.782	0.771	/
Two steps	0.762	0.559	0.645	0.240

Table 3.13: Overall F-measures estimation for two steps filtering

A sample of relation candidates after the two steps of filtering procedure is illustrated in Figure 3.7. Each candidate is presented in the format: $E1$; $Cmid$; $E2$; ($Cpost$). The nature of these relations is very diverse, with a certain redundancy for the $Cmid$ part for frequent

[14]NB: Precision is calculated by comparing the finally extracted relation instances with the reference, so the overall precision is the same with the precision of statistical filtering.

relations. As stated earlier, $Cmid$ bears most of the meaning of the relation instances in most of the cases, with $Cpost$ providing complementary information. Nevertheless, $Cpost$ sometimes plays the most important role in a relation, such as the last example: the "CEO" in $Cpost$ brings more information than the "step in" in $Cmid$.

With all extracted relation instances of such heterogeneity, an important thing to do is to organize them properly. Semantic and thematic clustering of relations will be discussed in the next chapter.

3.8 Conclusions and Perspectives

In this chapter, a relation prototype is first of all defined by focusing on binary relations between named entities. A processing pipeline is then proposed to extract and select relation instances from sentences that meet certain requirements. Relation extraction starts with very limited constraints to cover as many as possible relation instances, which means one verb between a named entity pair in a sentence. Following this initial extraction, two steps of filtering are applied to get rid of false relations, first step with filtering heuristics and second step with a statistical classifier.

Heuristics defined for relation filtering aim at eliminating false relation instances in large quantity efficiently. On one hand, heuristics remove a part of true relation instances but the ratio of negatively filtered relation candidates (true relations considered as false by heuristics) stays acceptable. On the other hand, the current heuristics are not sufficient to guarantee an ensemble of relation candidates with high enough quality. However, these heuristics can always be extended. For example, the verbs used for discourse elimination are actually limited to "say" and "present". An annotation of attribution words on the Penn Discourse TreeBank[15]) shows that attribution cues are dominantly expressed by verbs (96%), and among all these verbs, the verb "say" holds a proportion of 70% in the attribution database annotated. Other frequent words include "add", "note", "think", "believe", and etc. However, the presence of such a word does not indicate an attribution for certainty. The ratio of attribution expressed by the verb "say" is 0.60 and it is 0.43 for "add" (Pareti, 2012). Consequently, any additional words into discourse heuristic brings also the effect of deleting many potential true relations. More experimental evaluations should be conducted for heuristic qualification.

[15]Penn Discourse TreeBankPDTB is a large scale corpus annotated with information related to discourse structure and discourse semantics. Details are available at `http://www.seas.upenn.edu/~pdtb`.

John; starred in the backfield for; *Ohio State* ;
Bush; headed into the debate at; *University of Miami* ;
Kerry; told; *ABC News* ;
Ralph Nader; , deemed by the bipartisan; *Commission on Presidential Debates* ;
Bush; himself must make it clear to; *Congress* ;
Dennis Hastert; , told; *The Washington Post* ;
Mary Lou Wiegand; came into the; *Teamsters* ;
Adrienne Redd; , 43 , who teaches at; *Cabrini College* ;
Wagner; , who served in the; *Army* ; in the early 1980 's ;
Fox; , who spent seven seasons with the; *Lakers* ;
Derek Fisher; signed with; *Golden State*; ;
Alessandra Stanley; reviews two new television series on; *ABC* ;
Steve Reed; to give the Dodgers a 4-2 victory over the; *Colorado Rockies*; ;
Christopher Simmons; , are urging the; *Supreme Court* ;
Keith; told; *USA Weekend* ;
Brown; delivered rousing speeches to the; *Labor Party* ;
Johnny; told; *Rolling Stone* ; a few months ago
Dalglish; , who is the executive director of the; *Reporters Committee* ;
Tom Kelly; , a former trainer who is in the; *Hall of Fame* ; and is Pat 's father
Matt Jones; , lead the; *SEC* ; in scoring (39.8 points a game) and are averaging 477.5 yards of offense a game .
R-Fla.; , commenting on his home network; *MSNBC* ;
Bill Adler; , a writer and producer who worked with; *Simmons* ;
Bailey; worked at; *McDonald 's* ;
Sonia Murray; writes for; *The Atlanta Journal-Constitution* ;
Thomas Jones; leads the; *NFC* ; in rushing with 329 yards .
Aida Alvarez; , who ran the; *Small Business Administration*; during the Clinton presidency .
Smith; was a vice president at; *DHL Airways*; and previously worked at other airlines .
Gerald Grinstein ; stepped in as; *Delta*; 's CEO on Jan . 1 .
...

Figure 3.7: Examples of relation instances for the category PERSON-ORGANIZATION

Nevertheless, the objective here is not to find optimal heuristics but to offer an operational method for the second step of filtering with machine learning models. The current heuristics succeed in doing so by facilitating the relation validity annotation and a better balanced corpus for statistical training.

Filtering with statistical models was experimented with different ways of representing relations and with different types of statistical models. CRF models with sequential representation of relations achieve the best results. The performance of the trained classifiers was also compared favorably to existing systems.

The filtering procedure proves to be capable of removing false relation instances with a satisfying performance of recall and precision, which provides an large ensemble of more reliable relation candidates for the relation clustering in the next chapter. In addition to the evaluation of the filtering quality, the impact of the filtering procedure on relation clustering will be analyzed in Chapter 5.

Chapter 4

Relation Clustering

Once relation candidates are extracted and filtered, an important work to do is to organize them properly in order to provide the end-user with a more concise and understandable information. We discuss in this chapter the difficulties of clustering relation instances and present a multi-level clustering method. We present first different similarity measures between words and different clustering algorithms. The multi-level clustering method for semantically grouping relation instances is then detailed: a basic clustering is performed as a first step to group similar linguistic expressions with the purpose of forming precise basic clusters; a semantic clustering is then performed as a second step to group semantically similar basic clusters so that larger semantic clusters are formed. At last, a topic-based clustering method is proposed to integrate thematic information into relation clusters.

Contents

4.1 Relation Clustering Problematic 68

4.2 Similarity Measures for Clustering 73

4.3 Clustering Algorithms 78

4.4 Basic Clustering 85

4.5 Semantic Clustering 90

4.6 Topic-based Relation Clustering 94

4.7 A Summary of Our Clustering Approaches 98

4.1 Relation Clustering Problematic

4.1.1 Difficulties of Relation Clustering

Although unsupervised relation extraction has been gaining attention in recent years, most of the researches in this field concentrate on the relation extraction step and less work has been proposed on the semantic organization of the relation instances after they have been extracted. Most of the emphasis was put on how interesting relation candidates can be discovered and extracted.

For example, TEXTRUNNER (Banko et al., 2007) is capable of retrieving relations tuples from millions of Web pages. However, since relations are of such diversity and quantity in open domain, it is very important to properly organize all extracted relation instances in order to present the results to an end-user. In particular, similar expressions or synonymous phrases should be grouped together. Or going even further, more general paraphrase phenomena should be dealt with. In the case of TEXTRUNNER, extracted relation instances are indexed for querying, which makes these relation instances more accessible to users. However, TEXTRUNNER creates no semantic organization for their extracted instances.

The quantity of relation instances and the diversity of relation types are the advantages of unsupervised relation extraction in open domain. At the same time, these advantages are also the difficulties to solve for clustering relation instances.

Quantity issue The first difficulty for relation clustering is the scalability of the clustering process. Most of the clustering algorithms start from a similarity matrix that can be difficult to compute when the number of items to be clustered is large. This problem is bypassed in some researches by limiting the number of relation instances, either directly by a maximum number or through the limited initial size of documents. For instance, experiments in Rozenfeld and Feldman (2007) chose to process only for a maximum of 4,000 relation instances while Yan et al. (2009) employed their approaches restrictively to 526 Wikipedia documents. However, our objective is to tackle all extracted relation instances, the volume of which is 165,708 in our experiments.

Diversity issue The objective of clustering is the detection of homogeneity among heterogeneity while we need to deal with the heterogeneous nature of relations in massive data set. The first kind of diversity is that a corpus in open domain may contain abundant unknown relation types so that the cluster number can not be predicted in advance. One way to overcome this problem is to fix or to evaluate *a priori* the number of clusters to build.

For instance, Yan et al. (2009) chose to set arbitrarily the number of clusters according to the document set; Rozenfeld and Feldman (2007) tested different cluster number values for the same task; Gonzàlez and Turmo (2009) adopted the Akaike Information Criterion to evaluate this number in one of their experiments. We choose to use clustering methods that make no pre-assumption on the number of clusters.

The second kind of diversity is the variety of relation expressions that can exist for the same or similar relation types. Similar relations can be expressed by the same words or synonymous words in different linguistic forms. Clustering approaches should be capable of dealing with these different linguistic phenomena.

When diversity meets quantity For synonymous words or even more complicated paraphrase phenomena, complex semantic similarity measures can be used for calculating similarity values. However, complex semantic measures are often time-consuming for computation and this computation becomes more laborious when tens of thousands relation instances need to be treated. Therefore, the proposed approach should deal with the diversity issue and the quantity issue in the same time. The relation clustering system we propose to tackle these issues is detailed in the next sections.

4.1.2 Relation Clustering System Design

Our strategy is to first separate relation extraction and relation clustering into two individual tasks processed one after another. In systems such as (Hasegawa et al., 2004), the clustering method plays a dual role since relation instances are clustered and extracted in the same time. Extracted relation instances are those with a high score in relation clusters. But there are two limits to this kind of approach. On one hand, the clustering procedure is relatively time-consuming at large scale and is only feasible when a certain quantity of instance examples is accumulated, which limits both the scalability and applicability of the relation extraction procedure. On the other hand, without an attentive prior candidate selections, clustering procedure has to take into account all relation instances, including false relations. The separation of these two tasks makes the relation extraction procedure more scalable and the similarity computation for relation clustering procedure less heavy. The filtering procedure in our relation extraction system removes large amount of false relations so that the clustering procedure can concentrate on the selected instances (the number of relation candidates decreases from 677,375 to 165,708). Based on the remaining relation instances, a multi-level relation clustering is then proposed, the results of which are then integrated with a context clustering for a topic-based relation clustering.

Multi-level Relation Clustering

Arguments of relation instances in our system are named entities so that each relation instance is first of all characterized by its pair of named entity types. Hence, relation instances are first classified into different categories according to this pair of named entity types. Relation clustering methods are then applied inside each relation category.

Similar relation instances in each relation category can then be grouped. Here, two kinds of similarities must be distinguished: a semantic relation expressed with the same words and a semantic relation expressed with different but semantically similar words. The first kind of similarity refers to different linguistic forms using the same words for expressing the meaning of the relation, such as the example for the relation based on the word *create* in the category PER-ORG:

{ *create, create the, that create, who create the, etc* }

Similarity values among these relation instances can be calculated with a direct phrase comparison using, for instance, a cosine measure applied on vectors of each relation instance represented as bags of words. The second kind of similarity requires more complex semantic measures to detect for instance the similarities between the following words:

{ *create, establish, found, launch, inaugurate, etc* }

WordNet-based measures or distributional similarity measures can be adopted in this case. However, they are generally more time-consuming for computation.

A multi-level clustering method is proposed as shown in Figure 4.1 to treat these two types of phenomena successively. First, a basic clustering groups relation instances with very similar expressions. Using simple measures, such basic clustering is efficient for grouping large amount of expression variations. Basic clusters of high precision can thus be formed by setting a relatively high threshold of the similarity value. Then, a semantic clustering groups similar basic clusters together based on more complex semantic similarities among these basic clusters.

There are several reasons for the design of multi-level clustering. First of all, since relation types are unknown in unsupervised information extraction, basic clustering provides a way to discover the potential *labels* for relations. Second, simple similarity measures are more efficient than complex semantic similarity measures. The multi-level clustering reduces the calculation of semantic similarities from all pairs of relation instances to all pairs of basic clusters. Moreover, the redundant information inside each basic cluster makes

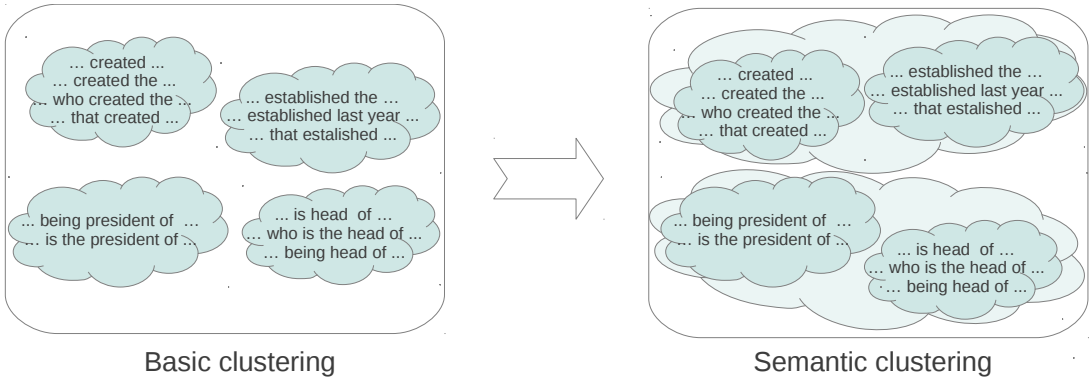


Figure 4.1: The procedure of multi-level relation clustering: basic clustering and semantic clustering

the semantic clustering of the second level much more effective since the similarities between basic clusters of the first level is less sensitive to outliers than similarities between all individual instances (Cheu et al., 2004).

Context Clustering and Topic-based Relation Clustering

We extract relation instances at the sentence level. However, they appear in a wider context and the topic information of this context can also be useful to characterize these relation instances. To get advantage of this topic information, each document of the corpus is thematically segmented. Each resulting thematic segment (C_i) includes an ensemble of neighbouring sentences which may include multiple relation instances (R_{ij}) and this thematic segment is considered as the context for these relation instances. We then applied a context clustering procedure which groups similar thematic segments to form context clusters so that each context cluster refers to a specific topic as shown in Figure 4.2.

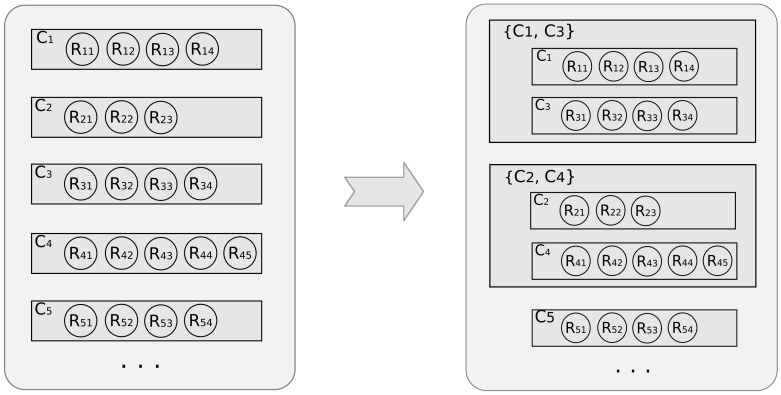


Figure 4.2: The procedure of context clustering

Topic-based relation clustering is then performed by combining this context clustering and the previous relation clustering with different integration strategies described in Section 4.6.

Overview of the Relation Clustering System

An overview of the relation clustering system is given in Figure 4.3. Basic clustering and semantic clustering are applied successively for each relation category while context clustering is independent of relation categories and can thus be applied directly on the linguistically processed corpus.

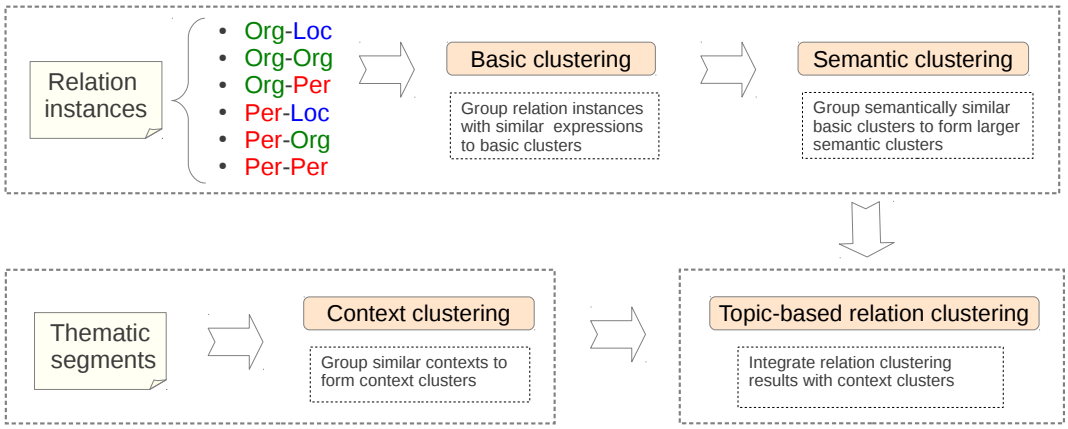


Figure 4.3: Overview of relation clustering and context clustering

In the remaining part of this chapter, we first discuss the basic similarity measures and semantic similarity measures along with the similarity matrix calculation issues in Section 4.2. In Section 4.3, we present different clustering algorithms that are suitable for relation clustering in our case and also our choice of algorithms for the different clustering tasks. Details for basic clustering and semantic clustering will be given respectively in Section 4.4 and Section 4.5. At last, Section 4.6 presents the context clustering methods and topic-based relation clustering which integrates relation clustering and context clustering in different ways.

4.2 Similarity Measures for Clustering

4.2.1 Basic Similarity Measures

Notion of Similarity Metrics

In our clustering tasks, the objectives are to group relation instances or thematic segments, both using bag-of-word representation. For example, if the $Cmid$ part of relation instance is represented as a bag-of-words, the similarity between two relation instances can be measured as a similarity between two vectors of terms, in which the terms are the words in $Cmid$ part of relation instance. Therefore, we start the discussion by similarity measures between vectors.

More formally, as presented in (Theodoridis and Koutroumbas, 2009), given a data set X and two vectors $\mathbf{x}$, $\mathbf{y}$ in the data space, a *similarity measure* (Sim) on X is a function:

$$Sim : X \times X \rightarrow \mathcal{R}$$

such that

$$\exists Sim_0 \in \mathcal{R} : -\infty < Sim(\mathbf{x}, \mathbf{y}) \leq Sim_0 < +\infty, \quad \forall \mathbf{x}, \mathbf{y} \in X \tag{4.1}$$

$$Sim(\mathbf{x}, \mathbf{x}) = Sim_0, \quad \forall \mathbf{x} \in X \tag{4.2}$$

and

$$Sim(\mathbf{x}, \mathbf{y}) = Sim(\mathbf{y}, \mathbf{x}), \quad \forall \mathbf{x}, \mathbf{y} \in X \tag{4.3}$$

Where $\mathcal{R}$ is the set of real numbers.

If in addition

$$Sim(\mathbf{x}, \mathbf{y}) = Sim_0 \quad \text{if and only if} \quad \mathbf{x} = \mathbf{y} \tag{4.4}$$

and

$$Sim(\mathbf{x}, \mathbf{y})Sim(\mathbf{y}, \mathbf{z}) \leq [Sim(\mathbf{x}, \mathbf{y}) + Sim(\mathbf{y}, \mathbf{z})]Sim(\mathbf{x}, \mathbf{z}), \quad \forall \mathbf{x}, \mathbf{y}, \mathbf{z} \in X \tag{4.5}$$

Sim is a *metric similarity measure*. Equation 4.2 indicates that the maximum possible similarity between two vectors is obtained when they are identical. The symmetry requirement of by Formula 4.3 is important also in the sense of similarity computation since the similarity between two objects only needs to be calculated once instead of twice[1]. More restrict

[1]The All Pairs Similarity Search algorithm (Bayardo et al., 2007) that we use for calculating similarity matrix in Section 4.2.3 takes advantage of this similarity symmetry property in order to optimize the computation.

conditions are necessary for a *similarity measure* to be a metric. Equation 4.3 requires that the maximum of similarity value are reached only then two vectors are identical. Equation 4.5 is known as the *triangular inequality*

A *similarity measure* gives the degree of proximity (degree of distance for *distance measures*) of two objects for the clustering algorithms. Generally, these measures are normalized so that they take the values in the interval [0,1] to offer a common basis of interpretation. Classical measures include *Cosine Similarity*, *Edit Distance*, *Euclidean Distance*, *Jaccard Similarity Coefficient*, *Dice's Coefficient*, etc[2]. Two of these measures involved with our experiments will be detailed later for comparison.

Cosine Similarity

The *Cosine similarity* corresponds to the cosine of the angle between two vectors, computed from their dot product and magnitudes. It is one of the most used measures in high-dimensional spaces, such as in *information retrieval* or *text mining*. For example, given two phrases P_a and P_b, each being an ensemble of words:

$$P_a : W_1, W_2, W_3, \ldots, W_M$$
$$P_b : W_1, W_2, W_3, \ldots, W_N$$

$\mathbf{V}_{P_a}$ and $\mathbf{V}_{P_b}$ are the term vectors that stand for the bag-of-words representations of two phrases P_a and P_b. Therefore, the *Cosine Similarity* between two vectors $\mathbf{V}_{P_a}$ and $\mathbf{V}_{P_b}$ is given as:

$$Sim_{cosine}(\mathbf{V}_{P_a}, \mathbf{V}_{P_b}) = \frac{\mathbf{V}_{P_a} \cdot \mathbf{V}_{P_b}}{||\mathbf{V}_{P_a}|| \times ||\mathbf{V}_{P_b}||} \tag{4.6}$$

In the case where no particular weighting is adopted, the value of each term in these two vectors reflects directly the frequency of this term in each phrase. Commonly used weighting strategies, such as the td-idf weighting, are based on the frequency of each term, which can not be negative. Consequently, the *Cosine Similarity* is generally not negative itself, ranging from 0 to 1. One advantage of the *Cosine Similarity* is that it only relies on the terms shared between the two vectors for both dot product and magnitudes, which makes it efficient to compute. In addition, it is independent of the length of phrases since both phrases are normalized to term vectors by the total term set.

It should be mentioned that the *Cosine Similarity* does not always respect the *triangular inequality* law. It depends on how the similarity value is converted to a dissimilarity value

[2]Some of these measures are *distance measures*. However, *similarity measure* and *distance measure* can be transformed from one to the other.

(distance)[3]. However, *triangle inequality* is not strictly required for a *similarity measure* when it is used for clustering algorithms (Anderberg, 1973; Jain and Dubes, 1988).

Edit Distance

Alternatively, if we consider each phrase as a sequence of words rather than a vector, the measure *Edit Distance*, also known as the *Levenshtein Distance* (Levenshtein, 1966), is widely used as a distance measure. It corresponds to the minimum number of insertions, deletions or substitutions required to transform one sequence into the other. If $editDist(P_a, P_b)$ is the *Edit Distance* between two sequence of words P_a and P_b, this distance measure can be easily converted into a similarity measure, such as proposed in (Lin, 1998b):

$$Sim_{edit}(P_a, P_b) \; = \; \frac{1}{1 \; + editDist(P_a, P_b)} \tag{4.7}$$

Sim_{edit} takes the maximum value of 1 when P_a and P_b are identical; otherwise, it varies in the interval]0, 1[.

Globally, these basic similarity measures can be used for detecting the similarity of phrases based on their common words, without considering further phenomena such as synonyms. These kinds of similarities are independent of any dictionary or knowledge base and are relatively efficient for computation. They are also easy to interpret as well: the more common words shared by two phrases, the more likely they correspond to similar relations.

4.2.2 Semantic Similarities between Words

Semantic similarity measures are used to identify words that have similar meanings. For example, the verbs "build" and "construct" are often considered as similar ones according to a synonym dictionary. Such similarity is more generally characterized by a normalized similarity measure, with values typically between 0 and 1 and the maximum value reached when the two words are identical. Resources such as WordNet (Miller, 1995) are often used to generate such a similarity measure for words of the same category. Apart from using existing knowledge, such semantic similarities can be based on corpus statistics

[3]If we consider the reciprocal of *similarity measure* ($\frac{1}{Sim}$) as the distance function (Dist), equation 4.5 can be transformed into $Dist(\mathbf{x}, \mathbf{z}) \leq Dist(\mathbf{x}, \mathbf{y}) + Dist(\mathbf{y}, \mathbf{z})$, which is the original definition of *triangle inequality* for distance. *Cosine Similarity* is not a metric when its multiplicative inverse is considered as its corresponding distance measure.

and the analysis of the distribution of word co-occurrences. Both types of measures are experimented in this thesis and will be discussed below.

WordNet-based similarities

WordNet is an English lexical database which groups words into sets of synonyms named synsets, providing semantic relations between these synsets in the form of hierarchies (Miller, 1995). Various types of measures were proposed to compute similarities between synsets using this hierarchical structure. Standard measures in the literature include *Path, Leacock and Chodorow (Leacock and Chodorow, 1998), Wu and Palmer (Wu and Palmer, 1994), Resnik (Resnik, 1995), Lin (Lin, 1998b), Jiang and Conrath (Jiang and Conrath, 1997) etc.*

The most direct similarity measure is the *Path* measure, which is the inverse of the shortest path between two synsets using node-counting. When the two synsets are identical, it reaches the shortest path and corresponds to the maximum similarity value. Starting from this idea, the *Wu and Palmer* measure considers the depth of two synsets (S_1 and S_2) in relation to the root in WordNet hierarchy tree and the depth of the least common subsumer[4] (S_0), with the following definition:

$$Sim_{wup}(S_1, S_2) = \frac{2 * depth(S_0)}{depth(S_1) + depth(S_2)} \tag{4.8}$$

Since $depth(S_1) \geq depth(S_0)$ and $depth(S_2) \geq depth(S_0)$, the value of Sim_{wup} varies between 0 and 1.

The *Resnik* measure includes in addition statistical information from a large corpus to compute the Information Content (IC) of the least common subsumer between the two synsets. The Information Content is defined by the probability of occurrence of the concept S_0 in a large corpus given by:

$$Sim_{res}(S_1, S_2) = IC(S_0) = -\log P(S_0) \tag{4.9}$$

The probability value $P(S_0)$ is necessarily between 0 and 1 so that the *Resnik* measure varies in the interval $[0, +\infty)$. The more frequent the concept S_0 is, the lower the similarity value will be.

The *Lin* measure makes *Resnik* measure easier to interpret by adding a normalization:

[4]The least common subsumer is the deepest common ancestor in the hierarchy shared by the two synsets.

$$Sim_{lin}(S_1, S_2) = \frac{2IC(S_0)}{IC(S_1) + IC(S_2)} \tag{4.10}$$

All the above similarities are given between synsets, each of which may contain several words. In the same way, each word may be included in different synsets. For word similarity calculation, a simple way of mapping synset similarity to word similarity is to choose the highest synset similarity among all possible synset pairs (Mihalcea et al., 2006).

Distributional similarities

Distributional similarities are based on the hypothesis that words occurring in the same context tend to have similar meanings, so that "a word is characterized by the company it keeps" (Firth, 1957). Therefore, given a large corpus, a vector of co-occurrences can be collected for each word (*context vector*). Different kinds of similarities can be computed depending on how distributional information is used to collect the co-occurrence vectors. Co-occurrences can be based on syntactic dependency relations (Lin, 1998a) or on a fixed size window of neighbouring words. In the last case, the size of the window is a parameter of the similarity. Small values tend to account for semantic similarities while larger values, that cover a wide text region, are more likely to account for topical similarities.

Once the context vectors have been built, the similarities between words are computed through the similarities between their context vectors, using standard measures between bag-of-words vectors such as the *Cosine Similarity*. Furthermore, each element in the context vector can have a more complex weighting such as *Pointwise Mutual Information* or $tf \cdot idf$. Different similarity measures and weighting configuration have been experimented on the AQUAINT-2 corpus in (Ferret, 2010).

4.2.3 Similarity Matrix Calculation

Several clustering algorithms, such as the ones that will be presented in Section 4.3, rely on a similarity matrix. These similarity matrix can be very costly to compute, in particular for large sets of relations like the one we want to process. Several tens of thousands of relation instances need to be processed in our case and the number of similarities is quadratic with respect to the number of relations instances.

In practice, all the pairs of objects do not need to be considered. A similarity search problem is the issue of finding only object pairs whose similarity is above a specific threshold, which make the similarity computation much more efficient. Approximation techniques are often applied when the data set is of large scale, which may result in a signifi-

cant amount of errors (Broder et al., 1997). However, we require an exact algorithm which is capable of searching all pair of similar relation instances between which the similarity is above a given threshold. *All Pairs Similarity Search* (Bayardo et al., 2007) meets our requirements for both the efficiency issue and exactitude issue, so that it was chosen the basic similarity matrix calculation.

All Pairs Similarity Search (APSS) is a parsimonious indexing approach combined with several optimizations that allows the computation to be efficient for similarity measures such as the *Cosine* measure. Given a set of vectors, an inverted list index of these vectors is built dynamically. The similarity search for each vector is done by treating this vector as a query to find the set of matching documents filtered by a minimum score (similarity threshold). The inverted index structure is also used for accumulating the partial similarity scores. Furthermore, several optimizations are added to this basic principle, such as the exploitation of the similarity threshold in order to reduce the amount of indexed data or the exploitation of a specific order of the data set. For binary vector data, where the component values can only be 1 or 0, a specific optimization is designed to strengthen the efficiency of the algorithm.

4.3 Clustering Algorithms

Clustering procedures aim at detecting similar objects and grouping them together. As discussed in (Jain and Dubes, 1988), a clustering procedure depends first on a representation for objects, for example in our case, the bag-of-words representation of the $Cmid$ part of relation instances. Then, it requires the definition of a similarity measure, as the *Cosine Similarity* or the *Edit Distance* presented in Section 4.2. At last, it needs the choice or the definition of a clustering algorithm to group objects according to their similarity.

The review of clustering algorithms made in (Jain et al., 1999) proposes a taxonomy of existing algorithms. Clustering algorithms have been developed considerably since then, whereas the general taxonomy is still valid. In general, clustering algorithms can be divided into two categories: hierarchical methods, which provide a series of partitions, and partitional ones, that give only one partition. Based on their taxonomy, different kinds of clustering algorithms are shown in Figure 4.4.

Hierarchical methods can be regarded as two types. Agglomerative approaches work in a “bottom up” way, so that each object starts in its own cluster and then clusters are merged as one moves up the hierarchy, while divisive approaches take a “top down” way, so that all objects start in one cluster and are then split into smaller clusters as one moves

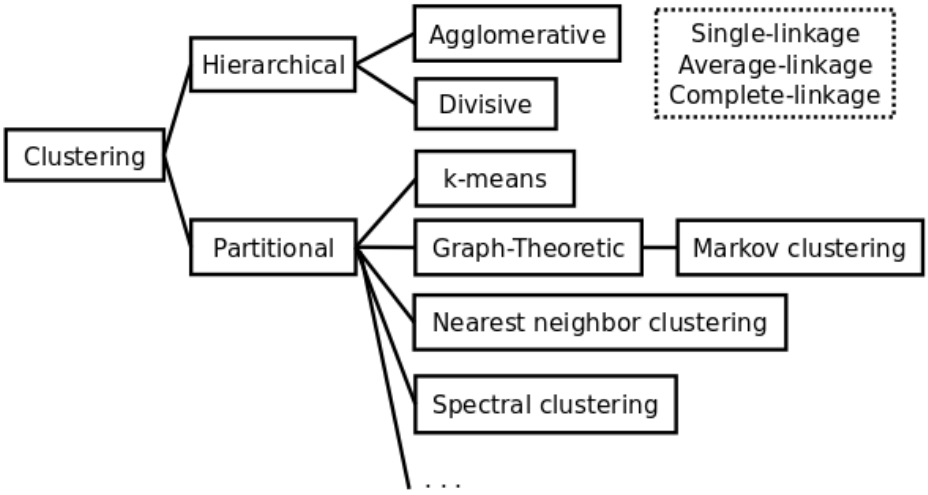


Figure 4.4: A taxonomy of clustering algorithms

down the hierarchy. During iterations of the merges for the agglomerative procedure, a linkage criterion is necessary for specifying the similarity between two sets of objects. Three criteria are often used: single linkage, average linkage, and complete linkage, which stand respectively for the minimal, average and maximal similarity value between two sets. In researches about relation clustering, complete linkage for hierarchical agglomerative clustering was first experimented in (Hasegawa et al., 2004) and (Rozenfeld and Feldman, 2006a). Single linkage was proved to be superior among these three linkages according to (Rozenfeld and Feldman, 2007). Nevertheless, hierarchical clustering is not very efficient for large data sets because of its complexity: $\mathcal{O}(N^3)$ for agglomerative clustering and $\mathcal{O}(2^N)$ for divisive clustering in terms of time complexity in a general case (Theodoridis and Koutroumbas, 2009)[5]. Moreover, the whole similarity matrix needs to be stored so that the space complexity is $\mathcal{O}(N^2)$ for both agglomerative and divisive clustering. With an optimized implementation, the time complexity of hierarchical agglomerative clustering can be reduced to $\mathcal{O}(N^2 log(N))$ (Kurita, 1991), which is still non negligible for massive data sets.

Partitional methods include various algorithms such as *K-means* (MacQueen, 1967), *Markov Clustering* (Van Dongen, 2000), *Nearest Neighbour Clustering* (Ertöz et al., 2003), *Spectral Clustering* (Luxburg, 2007). Most of the partitional algorithms are more efficient

[5]It has to be noted that N is the number of objects to be compared for similarities in the whole data set. For a more detailed complexity analysis, the vector size of each object should be considered as well. The complexity evaluations here concentrate only on the number of objects, which is the same case for all the time complexity or space complexity functions below.

than hierarchical ones from the viewpoint of computational complexity since no complete cluster hierarchy structures are attempted to be created.

K-means clustering aims at separating data sets into k clusters, where k is a predefined number. In the beginning of the clustering procedure, an ensemble of k random objects are chosen as the centroids of clusters and all objects are assigned to their nearest centroids selected previously. Then the centroid of each cluster is recomputed and objects are reassigned to the new centroids. This procedure is repeated until the convergence condition is met. In unsupervised information extraction, the cluster number k is difficult to evaluate. To overcome this problem, Chen et al. (2005) have adopted K-means clustering by automatically estimating cluster numbers with a stability-based criterion. The time complexity of K-means clustering is $\mathcal{O}(NkI)$, where I is the numbers of iterations and is generally significantly smaller than N (Theodoridis and Koutroumbas, 2009). If all objects and cluster centroids are stored, the space complexity is in the order of $\mathcal{O}(Nk)$. K-means algorithm is every efficient since it only needs to consider similarities between objects and the centroids of current clusters. However, the requirement of fixing *a priori* the number of clusters is an obstacle in open domain and the optimization needed to estimate this number leads again to a problem of complexity.

An appropriate clustering algorithm for unsupervised information extraction should have scalability properties to face huge document sets. Moreover, it is better to avoid the predefinition of the expected resulting number of clusters since this is not predictable for an unknown corpus in open domain. In the following of this section, more emphasis will be put on *Markov Clustering* and *Shared Nearest Neighbour Clustering*, which fulfill in a larger extent the requirements for relation clustering in unsupervised information extraction.

4.3.1 Markov Clustering

Markov Clustering (MCL) is a graph-theoretical clustering algorithm proposed in (Van Dongen, 2000). MCL algorithm performs the partitioning of a graph by the means of a series of random walks on the graph. In a similarity graph

$$\mathcal{G} = (\mathcal{V}, \mathcal{E})$$

where $\mathcal{V}$ and $\mathcal{E}$ are respectively the ensembles of vertices and edges, vertices are connected to each other by weighted edges that represent the similarity value between two vertices. A normalized similarity matrix M among vertices corresponding to the graph can then be

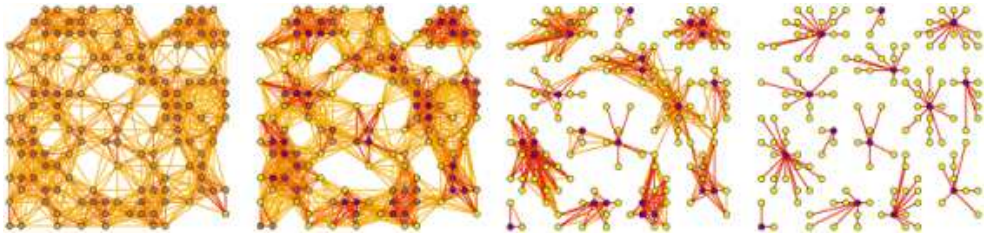


Figure 4.5: Markov Clustering procedure: evolution of edges (Van Dongen, 2000)

interpreted as the transition matrix for random walks in the graph. The MCL algorithm functions by applying two operations repeatedly: *expansion* and *inflation*.

- ***Expansion***

$$M = M * M \tag{4.11}$$

  Expansion operation implements a random walk of length 2 from one vertex to another, with the transition probability given by M. A length longer than 2 would be possible but would add computational time complexity for each iteration. This operation allows the flow to explore different regions of the graph.

- ***Inflation***

$$M_{i,j} = \frac{M_{i,j}^r}{\sum_i M_{i,j}^r} \tag{4.12}$$

  Inflation operation raises every value inside M with a power r and then normalizes each column to sum to 1. This operation strengthens the strong links between vertices and weakens the weak links, so that the inhomogeneity of each column is exaggerated. Usually, a power $r = 2$ is taken.

The *expansion* and *inflation* operations are repeated until the convergence condition is met. Intuitively, there are more links inside one cluster and less links between different clusters, which means that a random walk starting from one vertex is more likely to stay in the cluster containing this vertex than to go to another cluster. Therefore, the *expansion* operation discovers the clusters by detecting where the *flow* gathers. On the other hand, the *inflation* operation adds a non-linearity into random walk process to avoid each column of the transition matrix M from ending at principal eigenvector. Moreover, the *greedy* nature of *inflation* strengthens the flows inside a cluster and weakens the flows among different clusters. Therefore, after a series of random walks, strong links are gathered into separated clusters and weak links are eliminated, as illustrated in Figure 4.5.

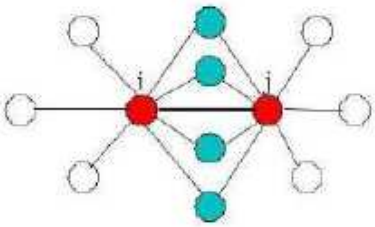


Figure 4.6: Shared neighbours between two objects

Theoretically, the MCL algorithm is rather time-consuming, with a time complexity of $\mathcal{O}(N^3)$. However, as presented in (Van Dongen, 2000), computation can be reduced by a *pruning* procedure, which can be an exact pruning by considering only the k largest entries of each column of the similarity matrix or a threshold that replaces all edges with a similarity below this threshold to zero. Since we use APSS for similarity calculation between relations, the threshold for APSS serves at the same time as the threshold for MCL graph *pruning*. The similarity matrix given by APSS is directly transformed into a similarity graph by associating each relation with a vertex and each non-zero similarity with a weighted edge between two vertices. The time complexity after *pruning* becomes $\mathcal{O}(NM)$, where M is the number of non-zero edges and M is often significantly smaller than N^2, the total number of vertices[6]. At the same time, the space complexity is reduced from $\mathcal{O}(N^2)$ to $\mathcal{O}(M)$.

4.3.2 Shared Nearest Neighbour Clustering

The basic principal of Shared Nearest Neighbour Clustering (SNN) is presented in (Jarvis and Patrick, 1973), assuming that the similarity between two objects can be measured by the nearest neighbours they share. SNN was applied to document clustering by (Ertöz et al., 2001, 2002, 2003) and used in the Natural Language Processing field for word sense induction (Ferret, 2004).

In the algorithm of (Jarvis and Patrick, 1973), a shared nearest neighbour graph is first constructed from the similarity matrix. A link between object i and j is created if and only if i and j have each other in their k nearest neighbour lists, as shown in Figure 4.6.

The weight of the link between object i and j can be directly the number of shared neighbours:

[6]For example, there are $N = 38,786$ relation instances for the relation type PER−PER after the filtering procedure so that a total of 1,504,353,796 (N^2) potential edges exist. With a basic cosine similarity applied to the $Cmid$ part of relation instances, a threshold 0.45 has the effect of reducing the number of non-zero edges to 436,801 (M), which is more than 3 orders of magnitude less than the original N^2.

$$weight(i,j) = size(L_i \cap L_j) \tag{4.13}$$

where L_i and L_j are the lists of k nearest neighbours of objects i and j.

Alternately, the weighting function can penalize far neighbours by taking into account the order of nearest neighbours, such as:

$$weight(i,j) = \sum_{v \in (L_i, L_j)} (k - p_{v,i} + 1) * (k - p_{v,j} + 1) \tag{4.14}$$

where k is a parameter fixing the number of considered nearest neighbours and $p_{v,i}$ and $p_{v,j}$ are respectively the positions of each shared neighbour in i or j's lists.

Ertöz et al. (2002) adopted Formula 4.14 for weight calculation. Once all these weighted links are created, a threshold is set to distinguish strong links from weak links. Ertöz et al. (2002) have also introduced core objects and noise removal to the algorithm. In fact, the objects are categorized into core objects or noise objects, according to their number of strong links. The core objects are used as seeds to initialize the clusters by associating the most similar objects to them, whereas the noise objects are regarded as outliers. Subsequently, objects not labelled as core or noise objects are associated with the nearest cluster.

One of the main advantages of SNN clustering, the same as the MCL algorithm, is that the approach does not require a prefixed number of clusters, which makes it more suitable for our unsupervised information extraction tasks. Moreover, SNN clustering does not focus on values but on densities, which make the clustering algorithm more independent from different scales of similarity measures.

The main drawback of SNN clustering concerns its high number of parameters, such as the size of the neighbourhood of a vertex or the percentages of strong links, core objects or noise. All these parameters will be detailed in the experiments of the next chapter.

SNN clustering requires $\mathcal{O}(N^2)$ time for computation if all pairwise similarities are computed to find the k nearest neighbours of each object. Since only the similarities with neighbours are stored, the space complexity remains at $\mathcal{O}(Nk)$. A further discussion about complexity analysis of SNN clustering can be found in (Ertöz et al., 2003).

4.3.3 Clustering Algorithm Choice

For a given task, the choice of a clustering algorithm should take into consideration the specific constraints of the task and the characteristics of the distribution of data. The first

consideration refers to the feasibility of the application of a clustering algorithm for a given task while the second refers to the suitability of a clustering algorithm on a data set to produce the best clustering results.

The first constraint of our task is that the clustering algorithms should be efficient enough to treat large size of data set. The second constraint is that the clustering algorithms should make no hypothesis about the number of clusters to generate since the number of relation types can not be known in advance in open domain. A synthesis of time complexity and space complexity is illustrated in Table 4.1 for clustering algorithms discussed earlier in this chapter[7]. Most of the hierarchical algorithms require heavy computations and K-means clustering needs a predefined number of clusters to initialize the cluster structure. As a result, we have concentrated on MCL algorithm and SNN clustering for dealing with our large sets of relation instances.

Clustering algorithms	**Time Complexity**	**Space Complexity**
Hierarchical - Agglomerative	$\mathcal{O}(N^3)$	$\mathcal{O}(N^2)$
Hierarchical - Divisive	$\mathcal{O}(2^N)$	$\mathcal{O}(N^2)$
K-means	$\mathcal{O}(Nkq)$	$\mathcal{O}(Nk)$
MCL	$\mathcal{O}(NM)$	$\mathcal{O}(M)$
SNN	$\mathcal{O}(N^2)$	$\mathcal{O}(Nk)$

Table 4.1: Time and space complexities for different clustering algorithms

MCL algorithm and SNN clustering do not work in the same way. Considering the characteristics of the data set, we investigate the most suitable of these two clustering algorithms for each of our three clustering tasks respectively: basic clustering, semantic clustering and thematic clustering.

MCL algorithm makes no assumption on the size of each cluster while it requires a threshold of similarity to ignore all unnecessary edges in the similarity matrix for the efficiency of random walks. SNN clustering can be very efficient even without a pruning threshold of similarity, but it is highly parameterized. More importantly, it needs to fix the number of nearest neighbours considered for similarity calculation, which can not be obviously determined in all cases.

In the basic clustering step, a simple similarity measure (*Cosine*) was adopted, and its value reflects intuitively the proportion of common words between two phrases. Therefore, the similarity threshold can be easily fixed. In the semantic clustering, setting a similarity

[7]The table gives theoretical complexity values. However, it is worth noting that different optimizations exist to make these algorithms more efficient as (Kurita, 1991) for hierarchical agglomerative clustering or the bisecting K-means approach (Steinbach et al., 2000) for hierarchical divisive clustering.

threshold is not obvious because various complex semantic similarity measures with different scales of values were adopted. On the other hand, the sizes of different clusters can be very diverse in open domain for the basic clustering task, depending on the frequency of words for different relations. On the contrary, for the semantic clustering task, the number of interesting neighbours is relatively easier to interpret, because it refers, to some extent, to the average number of synonymous words or paraphrases, which is more stable than the values of the semantic similarity measures.

Therefore, the MCL algorithm was finally chosen for basic clustering with given similarity thresholds and SNN clustering was applied for semantic clustering with given neighbour numbers. This choice was also verified by experiments: SNN results are much worse than Markov Clustering results for basic clustering while better for semantic clustering. For context clustering, the *Cosine* similarity measure was adopted and since the cluster sizes can be very diverse, as in the case of basic clustering, the MCL algorithm was chosen.

4.4 Basic Clustering

The unsupervised information extraction tasks have to face a huge number of relation instance in open domain, as shown in Table 3.1 of Chapter 3. Consequently, computing similarity for all pairs of relation instances directly with semantic similarities based on WordNet or a distributional thesaurus would have a very high cost. On the other hand, the study of extracted relation instances show that one relation can be expressed by the same *key word* in many different ways with relatively slight variations. Basic clustering aims at grouping efficiently and precisely relation instances which are expressed by similar linguistic forms with the same *key word*.

For instance, an inventory of the relations based on the verb "*retire*" for the relation category "Person-Organization" provides a list of expressions of the $Cmid$ part of the relation instance, as illustrated in Figure 4.7.

The *key word* that contributes mainly to the semantic meaning of the relation is the verb "retire" while other words gives complementary information about the modalities of the relation. Since paraphrase with synonyms are not considered at this step, it is not necessary to use sophisticated similarity measures. Basic similarity measures such as the *Edit Distance* or the *Cosine Similarity* are more efficient to calculate so that *basic clustering* can stay scalable and process large data sets.

Finally, we chose to apply the *Cosine Similarity* to a bag-of-words representation on the $Cmid$ part of relation instances since preliminary experiments had demonstrated the

Taylor **retire** from *General Electric* and use to pastor at ...
Don Badie , 68 , **retire** from *Acme Steel* in 1989 after 31 years ...
Cooper **retire** from the *Air Force* in 1970 as a colonel ...
Hassan who **retire** from *Iraqi Airway* in 1986 say his wife ...
Howard Zinn a historian **retire** from *Boston University* and an old nader friend ...
Alan Greenspan finally **retire** from the *Federal Reserve* will he leave behind any ...
Jensen who have **retire** from *Sonoma State University* where he found project censored ...
Jake Fiala a downhiller who **retire** from the *U.S. Ski Team* last spring and now sell ...
Thomas M Coughlin **retire** from his job as the second ranking executive at *Mart Store* he instruct a subordinate to ...
...

Figure 4.7: Examples of variations of the linguistic expression of the $Cmid$ part of relation instances based on the verb "**retire**"

superiority of the *Cosine Similarity* over the *Edit Distance* for a similar task (Campion et al., 2010). APSS was then used to calculate efficiently the similarities between all pairs of relation instances with a value higher than a given threshold. The choice of thresholds will be discussed in the next chapter.

4.4.1 Term Weighting Strategies

All the words of the $Cmid$ part of a relation instance do not have the same importance. As we can observe from the examples of Figure 4.7, the verb "retire" and the preposition "from" are essential for this relation while the words "who" or "a historian" are not part of the core expression of the relation and provide an expression variation. On the contrary, the age "68" or the comma "," are not related to the relation at all. Therefore, the term weighting strategy is important useful for relation clustering. Three kinds of weighting configurations are experimented in our cases: binary, tf-idf and POS-based weighting.

Binary configuration

Binary weighting is the basic weighting configuration in which all the words of the $Cmid$ part of relation instances are given the same weight, equal to 1, in our bag-of-words representation. Hence, the $Cmid$ part of each relation instance is represented as a binary vector.

This is a baseline we used in the primary experiments of our method since the specific optimizations of APSS for binary vector similarity computation is more efficient than for weighted vectors.

Tf-idf weighting

The tf-idf (term frequency-inverse document frequency) is a standard weighting strategy widely used in information retrieval that takes into account the importance of a word in a document and in the set of documents (a document is a relation instance in this case). A simple choice of tf is the raw frequency of the term inside the document such as Equation 4.15.

$$tf(t, d) = f(t, d) \tag{4.15}$$

The idf component measures whether the term is rare or common across all documents, which is calculated by the logarithm of the quotient of the total number of documents divided by the number of documents containing this term as in Equation 4.16.

$$idf(t, D) = log\frac{|D|}{|\{d \in D : t \in d\}| + 1} \tag{4.16}$$

where $|D|$ is the number of all documents D and $|\{d \in D : t \in d\}|$ is the number of documents containing the term t, with a "+1" smoothing term for avoiding division by zero. The tf-idf is given by the multiplication of tf and idf.

$$tf\text{-}idf(t, d, D) = tf(t, d) * idf(t, D) \tag{4.17}$$

POS categorization

The tf-idf weighting scheme measures the importance of a word according to its power of discrimination in a corpus. Words that are not frequent in the corpus but relatively frequent in a document gain more importance[8]. However, some words that are essential to the meaning of a relation can be given a small weight if they are too frequent in the whole corpus. For instance, the verb "buy" and the noun "father" are very common words hence have low idf values, but they do hold the key role in semantic relations such as *buy*(ORG-ORG), or *father_of*(PER-PER).

Nevertheless, an analysis of the part-of-speech types of words shows that POS types can be divided into several categories according to their importance for characterizing the

[8]The document is the relation instance here.

semantic meaning of a relation. More precisely, POS types can mainly be divided into four categories:

- **(A) Direct Contribution**: This category contains words that contribute directly to the meaning of a relation, which includes verbs, nouns, adjectives, prepositions and particles. An important weight must be given to these words.

- **(B) Indirect Contribution**: This category includes words that are not directly linked with the meaning of the relation but are relevant to the expression of a relation instance. These words are mainly but not only adverbs, pronouns and are given a medium weight.

- **(C) Complement Information**: Words in this category provide only complement information about relation instances. They are for instance proper nouns and interjections. A small weight is given to these words.

- **(D) Noise**: Words such as symbols, numbers, determiners, coordinating conjunctions or modal verbs are considered irrelevant for defining the meaning of relations and are removed from the similarity calculation.

Table 4.2 shows the list of the POS types for each of these categories, with the configuration of their weighting in the first column of the table. The list of POS types is based on the *Penn Treebank* tags[9]. For POS types that are not covered by these categories, a *default* category is set with a default weight.

4.4.2 Labeling and Refinement of Basic Clusters

Our basic clustering procedure groups relation instances with similar expressions into clusters very efficiently. We observed that basic clusters are very precise when the threshold for APSS is high enough and that characterizing words of relation instances in each basic cluster have generally a much higher frequency than the other words. Most of these basic clusters are characterized either by a verb (e.g. *founded* for "a group founded by", "which is founded by") or by a noun (e.g. *head* for "who is the head of", "becomes head of"). Hence, we consider the most frequent word (verb or noun) in each basic cluster as the label of this basic cluster[10].

[9]These tags are word level bracket labels in the documentation of "Bracketing Guidelines for Treebank II Style Penn Treebank Project", which is available at `http://bulba.sdsu.edu/jeanette/thesis/PennTags.html`

[10]This idea can be found in other works with a different objective: for instance, Hasegawa et al. (2004) use the most frequent common words in a cluster for labeling cluster.

Category	Part-of-Speech	
A	VB	Verb, base form
	VBD	Verb, past tense
	VBG	Verb, gerund or present participle
	VBN	Verb, past participle
	VBP	Verb, non-3rd person singular present
	VBZ	Verb, 3rd person singular present
	NN	Noun, singular or mass
	NNS	Noun, plural
	JJ	Adjective
	JJR	Adjective, comparative
	JJS	Adjective, superlative
	IN	Preposition or subordinating conjunction
	TO	to
	RP	Particles
B	RB	Adverb
	RBR	Adverb, comparative
	RBS	Adverb, superlative
	WDT	Wh-determiner
	WP	Wh-pronoun
	WP$	Possessive wh-pronoun
	WRB	Wh-adverb
	PDT	Predeterminer
	POS	Possessive ending
	PRP	Personal pronoun
	PRP$	Possessive pronoun
C	NNP	Proper noun, singular
	NNPS	Proper noun, plural
	UH	Interjection
D	SYM	Symbol
	CC	Coordinating conjunction
	CD	Cardinal number
	DT	Determiner
	MD	Modal
Default	others	all other types of part-of-speech

Table 4.2: Different categories of weighting by Part-of-Speech

We also observed that some relation instances with similar expressions were not grouped together. In some cases, the *key word* that characterizes the relation was not considered as the most important word of the relation instances while it can be identified as such after their clustering by its frequency. As a consequence, for limiting as much as possible the burden of the next step of semantic clustering, an additional step of clustering refinement was added to group the relation instances with similar expressions that are missed by the MCL algorithm. We chose for this step to merge the *basic clusters* that share the same label to form larger *basic clusters*.

4.5 Semantic Clustering

The basic clustering presented above fails to group relation instances that are semantically similar but have different linguistic expressions, such as in the examples of *a company based in* and *which is located in* for the relation category ORG–ORG, because synonyms are not taken into account. A complete semantic clustering should be able to group various types of paraphrases instead of connecting only phrases with the same words (*create, found, work, etc*). The purpose of the semantic clustering step is to tackle this problem by grouping basic clusters based on a more sophisticated semantic similarity measure. Since each basic cluster is rather precise and contains very homogeneous relation instances, it will take full advantage of the redundant information inside each basic cluster.

4.5.1 Similarity Measures for Semantic Clustering

Semantic clustering differs from basic clustering as it computes cluster-to-cluster similarities while basic clustering is based on instance-to-instance comparisons. Cumulative information provided by different relation instances in each basic cluster is essential to define the similarity measure between clusters. At least three levels of similarities have to be investigated: similarities between words, between relation instances (represented by a phrase or a sentence) and between basic clusters.

Word-level similarity

The similarity measure of basic clustering is binary and only considers whether two words are identical or not. However, a semantic similarity $S_{W_{i,j}}$ between two words W_i and W_j much characterize to what extent they are synonymous, which can be implemented by adopting either WordNet-based or distributional similarities as discussed in Section 4.2.2.

Phrase-level similarity

This problem of identifying similarity at phrase level is more related to paraphrase recognition. An intuitive way of computing similarities between two phrases is to take the average of the word-level similarities between all possible word pairs for the considered phrases (which are the $Cmid$ parts of relation instances in our case). Given two phrases, P_a with M words and P_b with N words, represented as bag-of-words:

$$P_a : W_1, W_2, \ldots, \ldots, W_M$$
$$P_b : W_1, W_2, \ldots, \ldots, W_N$$

the phrase-level similarity is then defined by:

$$S_{P_{a,b}} = \frac{1}{\sum_{\substack{i\in[1,M]\\ j\in[1,N]}} w_i \cdot w_j} \sum_{\substack{i\in[1,M]\\ j\in[1,N]}} S_{W_{i,j}} \cdot w_i \cdot w_j \tag{4.18}$$

where w_i or w_j are the weights given to each words, which can be for example a generic tf-idf weight.

However, all word pairings do not have the same relevance, especially for two words that are not important in the expression of the relation. Another option is to match each word in one phrase only with the most similar word in the other phrase. Hence, only the most similar matches are taken into account rather than all pairs of words (Mihalcea et al., 2006). In this case, the similarity is not symmetric ($sim(P_a, P_b) \neq sim(P_b, P_a)$), since W_i being the most similar word in P_a with W_j in P_b does not guarantee W_j to be the most similar word in P_b for W_i. Therefore, the average of similarities in both direction is taken to make this measure a metric. This similarity is then defined formally by:

$$\begin{aligned} S_{P_{a,b}} = \frac{1}{2}\Big(&\frac{1}{\sum_{i\in[1,M]} w_i} \sum_{i\in[1,M]} \max_{j\in[1,N]} \{S_{W_{i,j}}\} \cdot w_i + \\ &\frac{1}{\sum_{j\in[1,N]} w_j} \sum_{j\in[1,N]} \max_{i\in[1,M]} \{S_{W_{i,j}}\} \cdot w_j\Big) \end{aligned} \tag{4.19}$$

Cluster-level similarity

Each basic cluster contains two or more relation instances. A complete-linkage or average-linkage between clusters makes the calculation very heavy because similarities between all phrase pairs of the two clusters are to be computed. On the other hand, the similarity value

will be biased if only one relation instance is randomly chosen as a representative of one basic cluster, even with the high precision of each basic cluster. Moreover, the estimation of a medium phrase for a cluster is not always obvious and may result in an important loss of information for each basic cluster while the redundancy of information is useful for make the similarity computation less sensitive to outliers.

The solution we propose is to merge the bag-of-word representation of all relation instances in a basic cluster to form a general bag-of-word representation of this basic cluster. Each word in this new bag-of-word representation will be associated with its frequency in the basic cluster. The hypothesis is that the most relevant words with respect to the relation of the cluster will appear more frequently; thus higher weight will be given to them. The frequency of words in a basic cluster can be regarded as a weighting parameter in order to take advantage of the redundancy of information in basic clusters. The same formula as Formula 4.18 or Formula 4.19 can be used.

Given two basic clusters C_a and C_b, with their bag-of-word representations:

$$C_a : W_1 : f_1,\ W_2 : f_2,\ \ldots,\ W_M : f_M$$
$$C_b : W_1 : f_1,\ W_2 : f_2,\ \ldots,\ W_N : f_N$$

where W_i is a word of a phrase and f_i is the frequency of this word in the basic cluster, the cluster-level similarity is expressed by:

$$S_{C_{a,b}} = \frac{1}{\sum_{\substack{i\in[1,M]\\ j\in[1,N]}} f_i \cdot f_j} \sum_{\substack{i\in[1,M]\\ j\in[1,N]}} S_{W_{i,j}} \cdot f_i \cdot f_j \tag{4.20}$$

$$S_{C_{a,b}} = \frac{1}{2}\Big(\frac{1}{\sum_{i\in[1,M]} f_i} \sum_{i\in[1,M]} \max_{j\in[1,N]} \{S_{W_{i,j}}\} \cdot f_i + \frac{1}{\sum_{j\in[1,N]} f_j} \sum_{j\in[1,N]} \max_{i\in[1,M]} \{S_{W_{i,j}}\} \cdot f_j\Big) \tag{4.21}$$

The cluster-level similarity measures with Formula 4.20 and Formula 4.21 are directly derived from Formula 4.18 and Formula 4.19. There might be a frequency bias problem for the Formula 4.21, Indeed, each basic cluster contains words with different frequencies, such as in the two following clusters:

C_a : found:3 actor:3 {e.g. PER *an actor who found* ORG, ...}

C_b : study:9 actor:1 {e.g. PER *study at* ORG, PER *an actor study at* ORG, ...}

In this example, C_a and C_b are not semantically similar basic clusters. However, the similarity from C_a to C_b is high because of the shared word *actor* and its relatively high frequency inside the first cluster. Even though the inverse similarity (from C_b to C_a) is lower, the average of both is influenced by the first one and stays relatively high. To solve this frequency bias problem, the frequencies of both matched words in both clusters are taken into account for the calculation of similarity in each direction, which turns Formula 4.21 into:

$$S_{C_{a,b}} = \frac{1}{2}\{\frac{1}{\sum_{i\in[1,M]} f_i \cdot f_j} \sum_{i\in[1,M]} \max_{j\in[1,N]}\{S_{W_{i,j}}\} \cdot f_i \cdot f_j + \frac{1}{\sum_{j\in[1,N]} f_i \cdot f_j} \sum_{j\in[1,N]} \max_{i\in[1,M]}\{S_{W_{i,j}}\} \cdot f_i \cdot f_j\} \quad (4.22)$$

This cluster-level similarity measure succeeds at taking into account the important information of each basic cluster for similarity without adding too much unnecessary similarity computation.

4.5.2 Part-of-Speech Issues and Similarity Choice

In general, the important words characterizing a relation are observed to be verbs and nouns in the $Cmid$ part of relation instances[11]. For semantic similarity measures, words within the same part-of-speech type are first compared, with the objective of grouping relation instances that are either mainly characterized by verbs as in:

$\{$*ORG found by PER*$\}$, $\{$*ORG establish by PER*$\}$

or mainly characterized by nouns as in:

$\{$*ORG be partner of ORG*$\}$, $\{$*ORG have cooperation with ORG*$\}$

In practice, we observed that, for WordNet-based measures, the Sim_{wup} similarity performs well with Noun-Noun comparisons, while the Sim_{lin} similarity performs better for Verb-Verb comparisons. For distributional similarities, both syntax-based and window-based similarities were tested for both Verb-Verb and Noun-Noun comparisons. As verbs are the more frequent than nouns, experiments concentrate first only on verbs, and were then extended to nouns and even adjectives in a second step.

Similar relation instances expressed by both verbs and nouns also exist in the corpus, such as:

[11] In addition, we observed that verbs are much more frequent than nouns as the most important words for the relations. This can be a bias caused by the way our relation instances were initially extracted as the presence of at least one verb between two entities is required

{*ORG cooperate with ORG*}, {*ORG have cooperation with ORG*}

Cross-category similarity was also considered using distributional thesaurus but did not produce an obvious improvement of performance, so that our semantic clustering algorithms were performed mainly based on similarities computed with words of the same category (verbs or nouns).

4.6 Topic-based Relation Clustering

The objective of relation clustering is to organize relation instances for characterizing their types, which can be useful for end users or for another information extraction system. Semantic clustering is based on the semantic meaning of relation instances, which is directly linked with the way these relation instances are expressed, more specifically the $Cmid$ parts of the mentions. However, the mention of a relation instance only provides information at a local level, more precisely at the intra-sentential level. From a larger point of view, each relation instance also belongs to a certain *context*, which refers to general themes such as politics, sports, economics, and etc.

The thematic information associated with each relation instance offers several possibilities by taking into account more global elements. First, thematic information makes it possible to organize relation instances into different themes, which is interesting from an applicative point of view for end users. Furthermore, thematic information can be useful in many ways to improve semantic clustering. One of our *a priori* hypothesis is that inside the same theme, relation instances are more likely to be semantically similar, so that semantic clusters can be formed in a more precise way. Another assumption is about the distinction of polysemous words. In fact, semantic clustering succeeds at grouping similar relation instances together, whereas some words may hold multiple meanings, which vary according to the context of words. This can lead to group relation instances that are not actually similar to each other semantically because the meaning of a word in a relation instance does not correspond to the meaning of the same word in another relation instance. Thematic information potentially gives a way to distinguish one meaning from another for relations in the same semantic cluster.

Consequently, we propose a topic-based relation clustering to add thematic information into our previously introduced semantic clustering methods. First of all, a *topic segmentation* method is used to generate contextual words for each relation instance. Context clustering is then applied to group these contexts according to their similarities so that each context cluster contains an ensemble of contexts and represent one specific theme. Differ-

ent methods of combining context clustering with semantic clustering are then explored in Section 4.6.2.

4.6.1 Topic Segmentation and Context Clustering

Given a set of documents, the theme of each document can be represented as a distribution of characteristic words (Blei et al., 2003), the problem of which is known as *topic identification*. From a more extensive view, each document can contain different themes or sub-themes, with each sub-theme covering one or several neighbouring sentences. The division of one document into multiple thematic segments corresponds to the well-known problem, called *topic segmentation* (Galley et al., 2003). Researches also show that topic identification and topic segmentation can be tackled together. For example, Ferret (2007) used the topics unsupervised discovered from a document to improve its segmentation into topical segments. On the other hand, topical segments can also be useful for topic identification. In our case, we have used a context clustering method for grouping similar thematic segments together to identify the theme of each segment.

Topic segmentation algorithm As stated in Section 3.3 of Chapter 3, each extraction relation instance follows a prototype made of three main parts in the defined relation prototype: a pair of named entities as arguments, the linguistic form of its mention and a thematic segment as its context. This segment comes from the application of a topic segmentation procedure to all documents during the *linguistic preprocessing* step. Each document is divided into several segments so that each extracted relation instance belongs to one thematic segment. The content words of each thematic segment are regarded as the context of each relation instance.

For thematic segmentation, many systems were proposed in the literature including LCseg (Galley et al., 2003), TextTiling (Hearst, 1997), SeLeCT (Stokes et al., 2004), (Dias et al., 2007), TopicTiling (Riedl and Biemann, 2012), (Guinaudeau et al., 2012), etc. Most of the systems contain two essential elements: a similarity measure between term vectors of segments and a strategy of segment boundary identification.

LCseg was finally chosen and applied on the AQUAINT-2 corpus because of its stable performance across various corpora. The principal hypothesis of LCseg is that major topic shifts are likely to occur when strong term repetitions start and end. More in details, the input documents are first preprocessed, keeping only their content words. Then, a sliding fixed-size window moves all over each document to segment and at each sentence bound, a measure is computed for evaluating the similarity of the two sides of the window. This

measure is based on the overlaps of lexical chains between the two parts of the sliding window. Hence, a profile of similarity is built at the document scale: a high value should be obtained for thematically homogeneous parts while a low value should be observed for parts in which a topic shift occurs. Therefore, a boundary between two thematic segments is identified each time the similarity value reaches a minimum point.

Context clustering algorithm Each thematic segment is an ensemble of adjacent sentences in a document. It is characterized by a list of content words which represents the context of a relation instance. The general idea for grouping these contexts is the same as the one underlying basic clustering. First, each context is turned into a vector by adopting a bag-of-word representation. APSS is then applied to compute the cosine similarities among these vectors, either using a binary vector or by weighting each word with its tf-idf value. Based on the resulting similarity matrix, these thematic segments are clustered with the MCL algorithm.

4.6.2 Combination of Relation Clustering and Context Clustering

Each context cluster contains an ensemble of contexts that are thematically similar. Since each thematic segment may contain one or more relation instances, the context clustering provides also a kind of thematic organization of extracted relation instances. Two different ways of combining context clustering with semantic clustering were experimented. The first combination consists in applying the two types of clustering successively, with one clustering based on the results of the other. The second way is to apply the two types of clustering in parallel and then to merge the two kinds of resulting clusters.

Sequential application of relation clustering and context clustering

We started by applying context clustering and relation clustering sequentially. Given an ensemble of relation instances with various contexts as shown in Figure 4.8, the first option (option 1 of Figure 4.8) is to first apply the context clustering on all thematic segments so that context clusters are formed, with each cluster containing a list of contexts along with their corresponding relation instances. The relation clustering is then applied for all relation instances in each context cluster. This option is designed to verify whether more precise relation clusters can be formed for relation instances inside similar themes.

Another option (option 2 of Figure 4.8) is to first obtain relation clusters by our relation clustering process. Context clustering is then applied for all thematic segments in each

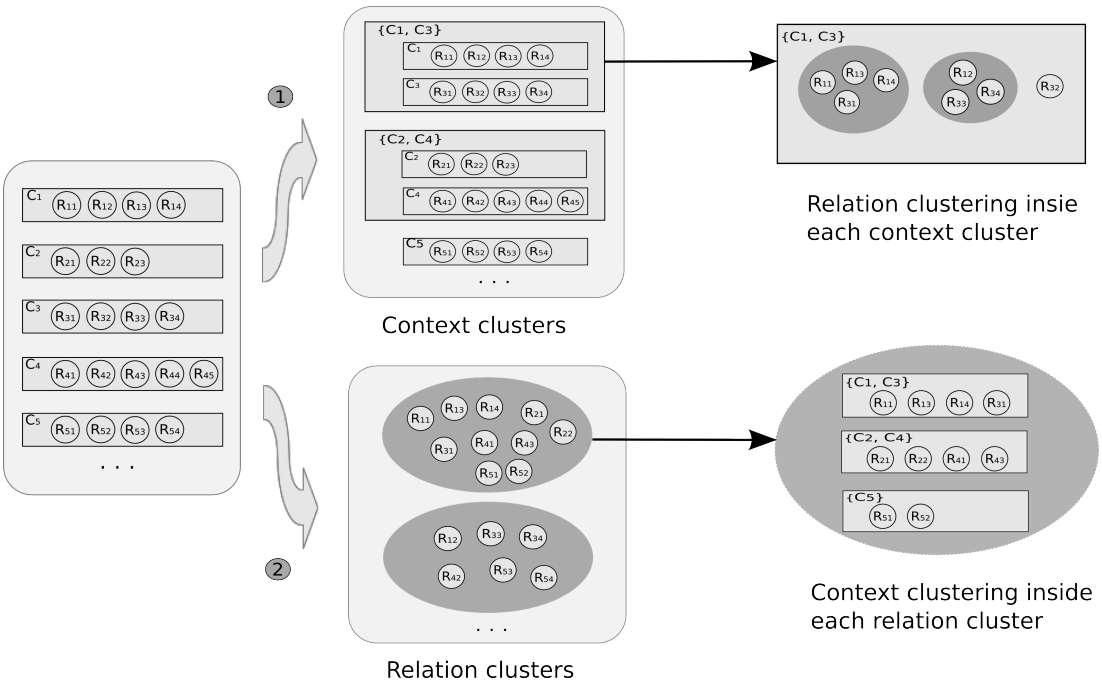


Figure 4.8: Topic-based relation clustering: the application of one clustering after another

relation cluster so that relation instances in each relation cluster is further grouped by their themes. The purpose of this option is to check whether different semantic meanings in one relation cluster can be differentiated by the themes of relation instances.

Both options achieve the objective of forming more precise relation clusters when compared to a reference. However, the experimental results, presented in more details in the next chapter, show that this increase of precision is associated with a significant drop of recall (a recall less than 0.01 for certain cases) so that the effect of the use of thematic information is difficult to analyze.

Integration of relation clusters and context clusters

When context clustering is applied first, generating many context clusters of various sizes, there are not always enough relation instances in each context cluster to form relation clusters of good quality. A similar problem is observed when context clustering is performed inside each relation cluster. Nevertheless, context clusters and relation clusters can also be obtained in parallel and independently, as in the first steps of two options of Figure 4.8. Then, rather than making a second step of clustering inside the first clusters, we can merge the two kinds of clustering results.

As shown in Figure 4.9, we look at each relation cluster to find out relation instances that are in the same theme (i.e. in the same context cluster). If two or more relation

instances belong to the same theme, they are pulled out to create a new relation cluster. All the isolated relation instances (i.e. no other relation instance can be found in the same theme) are left in the initial relation cluster. Details of experiments and their evaluations will be presented in the next chapter.

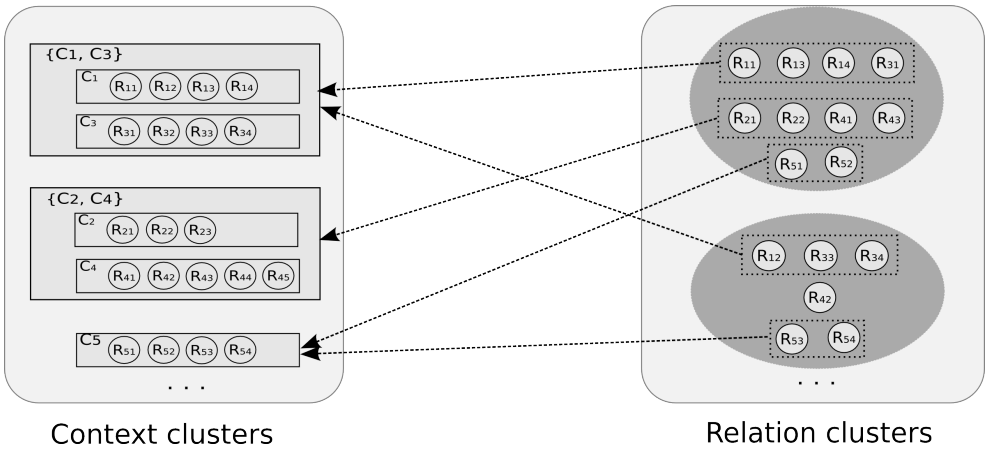


Figure 4.9: Topic-based relation clustering: the integration of two kinds of clusters

4.7 A Summary of Our Clustering Approaches

This chapter has presented our approaches for organizing relation instances in the framework of unsupervised information extraction. Based on our observations on a large corpus, we have distinguished relation instances that expressed with the same words from relation instances that are expressed with synonymous words. Hence, we have proposed a multi-level relation clustering method by first grouping relation instances which have similar linguistic expressions to build basic clusters in an efficient and precise way and then grouping basic clusters that share the same semantic meanings to form larger semantic clusters. In the basic clustering step, different weighting strategies of similarity computation have been presented. In the semantic clustering step, similarity measures at word level, relation instance level and basic cluster level have been analyzed. We have also discussed suitable clustering algorithms for our different clustering tasks.

Finally, we have presented a topic-based relation clustering with the objective of integrating thematic information into relation clustering. The thematic segments of relation instances are used to form context clusters so that each context cluster represents a specific theme. Different ways of integrating this context clustering into relation clustering have been proposed.

Chapter 5

Evaluations and Results

Evaluation of clustering is always a difficult task for unsupervised information extraction. In this chapter, we first discuss the general problems of clustering evaluation and then present our evaluation framework using both internal and external evaluation measures. To apply external measures, a reference is needed and we present how we built this reference efficiently in an iterative way. Within this evaluation framework, the impact of filtering procedure on relation clustering is first evaluated. Following evaluations include different experiments of basic clustering, semantic clustering and topic-based relation clustering. Result clusters are also illustrated for certain experiments.

Contents

5.1 Clustering Evaluation Problems 100

5.2 Clustering Evaluation Framework 102

5.3 The Impact of Filtering Procedure on Relation Clustering 114

5.4 Experiments of Basic Clustering 119

5.5 Experiments of Semantic Clustering 125

5.6 Experiments of Topic-based Relation Clustering 138

5.7 Conclusions and Perspectives 144

5.1 Clustering Evaluation Problems

Clustering methods are often adopted for unsupervised information extraction either as a way of discovering frequent relations or of grouping semantic relations (detailed in Chapter 2). The development of new clustering methods is an active field and approaches for the evaluation of clustering results have been well discussed in the literature (Manning et al., 2008; Theodoridis and Koutroumbas, 2009). However, in the context of unsupervised information extraction in open domain, evaluation of result clusters is still a challenging problem since large reference data often does not exist.

Some researches use unsupervised information extraction as a source of improvement for "traditional" information extraction by extending the coverage of models learned from annotated corpora. In this perspective, unsupervised information extraction modules are indirectly evaluated through their impact on the information extraction system which they are part of, as in (Banko and Etzioni, 2008) or (Gonzàlez and Turmo, 2009). The viewpoint in this thesis is different since unsupervised information extraction is used as a means to draw a global picture of the relations between a set of target entities for technology watch purposes. Hence, we are interested in evaluating more directly the clusters of relation instances built by this kind of process.

As stated above, the absence of gold standard is the main obstacle for clustering evaluation. Consequently, this evaluation issue is tackled more precisely by addressing the two following issues:

- how to evaluate clustering results without any reference?
- how to build a reliable reference for a given corpus and use it for evaluation?

The first issue is new in the field of unsupervised information extraction. More globally, existing internal criteria that allow to establish to what extent the clusters obtained correspond to the similarity measures between objects (Halkidi et al., 2002). The hypothesis is that a better distribution of similarities in the object space has the tendency of generating better separated clusters.

The second issue arises from the analysis of existing work, which performs an actual evaluation of whether two relation instances in the same cluster belong to the same semantic relation. In (Hasegawa et al., 2004), one of the first work in this domain, they performed a *a posteriori* evaluation of clusters of relation instances by assigning manually to each of them the relation type corresponding to the majority of the relation instances contained by this cluster. Then, recall and precision measures were computed by counting pairs of

relation instances that were correctly grouped or not according to assigned types to clusters. However, such *a posteriori* approach faces two problems which are linked together: first, because of its cost, an evaluation cannot be done each time a new clustering system or an existing clustering system with different parameters is tested; second, the results of the evaluation of one system cannot be used for the evaluation of another one as the reference built from the first evaluation is biased by the first system. This difficulty could be overcome to some extent by using a pooling technique, as it is often done in information retrieval for the evaluation of search engines. However, pooling requires a large number of different systems, which is only possible in the context of an evaluation campaign, and is made more difficult in the case of clustering by the fact that results are not structured by a set of known queries and are therefore more difficult to compare.

Rozenfeld and Feldman (2007) adopted a different approach, more directly linked to our viewpoint about the application of external measures. First, they annotated manually a restricted set of 200 relation instances and then, computed the *Jaccard coefficient* between their result clusters and their reference clusters at the level of relation instance pairs. The size of their reference set was however small, which is a limit of this evaluation. Gonzàlez and Turmo (2009) used the same principle but adopted as reference the relation instances annotated in a corpus in the context of a supervised information extraction task, more precisely, the Relation Mention Detection task of the ACE (Automatic Content Extraction) evaluation (Doddington et al., 2004).

In our case, to obtain a large set of reference clusters for evaluating our result clusters of relation instances in different experiments, we built a web-based tool to query and annotate interesting relations from a corpus in an iterative way. More than 4,000 relation instances were annotated within 80 relation clusters in a short time with this procedure. This reference was then exploited for the application of external measures to the clusters of relation instances produced by our unsupervised relation extraction system.

In the following of this chapter, the evaluation framework is first presented in Section 5.2, with the discussion of different internal and external measures. The method we used to build our reference is also presented in the same section. An evaluation of the impact of the filtering in the relation extraction process is presented in Section 5.3. Different experiments of basic clustering and semantic clustering are then discussed in Section 5.4 and 5.5. At last, experiments of topic-based relation clustering are presented in Section 5.6.

5.2 Clustering Evaluation Framework

5.2.1 Internal Evaluation Measures

When no reference is available, clustering quality is usually evaluated by a manual inspection of a subset of clustering results, which is likely to be biased as the resulting clusters tend to influence annotators. Hence, we chose another approach in the field of unsupervised relation extraction through the use of internal criteria. Such criteria establish to what extent the clusters obtained are representative of the similarity values between relations (Halkidi et al., 2002).

Different internal measures were proposed to evaluate clustering results or estimate the *clustering tendency*[1]. Classical measures include *Dunn Index* family measures, *Davies-Bouldin Index*, *expected density*, *connectivity*, etc (Stein et al., 2003; Handl et al., 2005).

Among various internal measures for clustering evaluation, we first chose *expected density*, since it was proved to have the best and the more stable correlation with F-measure for document clustering, especially compared to the more widespread *Dunn Index* (Stein et al., 2003).

Our clustering algorithms are based on the similarity graph of different instances while the measure *expected density* is defined based on the density of this graph. Formally, given a weighted graph (V, E, w) with a node set V, an edge set E and a weight function w, it is called sparse if $|E| = \mathcal{O}(|V|)$ whereas it is called dense if $|E| = \mathcal{O}(|V|^2)$. Therefore, the density of a graph can be calculated from the equation $|E| = |V|^\theta$. The density θ of a graph is then defined as:

$$|w(G)| = |V|^\theta \Leftrightarrow \theta = \frac{ln(w(G))}{ln(|V|)} \tag{5.1}$$

with $w(G) := |V| + \sum_{e \in E} w(e)$, where the weight function w is defined by the similarity between relation instances in our case, and $|V|$ is the number of relation instances.

For a set of result clusters $C = \{C_i\}$ with $C_i = (V_i, E_i, w)$, a local density θ_i can be given for each cluster. The *Expected density* is then computed by combining the local and global graph density of clustering:

$$\rho = \sum_{i=1}^{|C|} \frac{|V_i|}{|V|} |V_i|^{\theta_i - \theta} \tag{5.2}$$

[1] *Clustering tendency* tries to determine if applying clustering is likely to produce interesting results or not. It can also refer to the estimation of the number of clusters before clustering.

$|V_i|$ is the size of each cluster formed, and $\frac{|V_i|}{|V|}$ intends to balance the difference of size of clusters. It is easy to notice that this measure can be influenced by the size of the corpus, as bigger $|V_i|$ value tends to be produced in a bigger corpus during clustering. For the adaption of this measure to situations where there is a significant difference of corpus size, as it is the case when comparing clustering results with or without the filtering procedure, we choose to loosen the base of the exponential function $|V_i|$ in $|V_i|^{\theta_i - \theta}$ to make this measure less dependent on the corpus size, since it is the density rather than the number of vertices that we are interested in. The adjusted definition is given:

$$\rho' = \sum_{i=1}^{|C|} \frac{|V_i|}{|V|} \frac{\theta_i}{\theta} \tag{5.3}$$

Each local graph represents the similarity graph of each result cluster, so a higher value of the measure ρ' corresponds to a better clustering quality.

Connectivity is another internal measure (Handl et al., 2005), which evaluates how many nearest neighbours are not clustered together. The measure offers an opposite point of view to *expected density* measure, since it starts from each local similarity graph (i.e. nearest neighbours for each object) and then checks the clustering quality for objects in each local graph (i.e. whether neighbours are grouped or not) while the *expected density* starts from each result cluster and then checks the quality of similarity graph for all objects inside each result cluster (i.e. local graph density). The *connectivity* measure is defined by:

$$c = \sum_{i=1}^{|V|} \sum_{j=1}^{p} x_{i,nn_i(j)} \tag{5.4}$$

where p denotes how many neighbours are taken into account, $nn_i(j)$ is the j^{th} nearest neighbour of i and $x_{i,nn_i(j)}$ equals to 0 if i and $nn_i(j)$ are in the same cluster and equals to 1 otherwise.

The lower the connectivity value is, the less nearest neighbours are cut by clustering algorithm, which corresponds to a better clustering quality. As shown by its formal definition, connectivity depends directly on the corpus size. To adapt this measure to our context, a fixed-size subset of the total corpus is selected randomly for evaluation.

5.2.2 External Evaluation Measures

Internal measures provide a way of evaluating the clustering results intrinsically when no reference is available while external measures offer more comprehensive evaluations by

comparing result clusters with reference clusters. External measures such as *Rand Index*, *F-measure*, *Purity* and *Normalized Mutual Information* have been well discussed in the literature (Manning et al., 2008). *Rand Index* and *F-measure* are used to evaluate results at instance level, by verifying whether every pair of relation instances is correctly grouped together according to the reference, while *Purity* and *Normalized Mutual Information* evaluates results at cluster level by comparing each result cluster with reference clusters.

Instance-level measures

Given reference clusters with N relation instances, all pairs of relation instances in result clusters can be compared to those in reference clusters. Thus, *Rand Index* can easily be defined to check how all $N(N-1)/2$ pairs of relation instances are grouped. A good clustering method should assign similar instances to the same cluster and dissimilar ones to different clusters. Hence, there are four kinds of decisions. First, a true positive (TP) decision assigns two similar relation instances to the same cluster while a true negative (TN) one assigns two dissimilar relation instances to different clusters. TP and TN are both correct decisions. On the other hand, there are two incorrect decisions: false positive (FP) decisions, which assign two dissimilar relation instances to the same cluster and false negative (FN) decisions, which assigns two similar relation instances to different clusters. The *Rand Index* measures the clustering accuracy, which is defined by:

$$Rand\ Index = \frac{TP + TN}{TP + FP + FN + TN} \tag{5.5}$$

F-measure can be defined at the same time, relying on the precision P and recall R calculated by:

$$P = \frac{TP}{TP + FP}\ ,\ R = \frac{TP}{TP + FN}\ ,\ F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R} \tag{5.6}$$

Generally, for a clustering evaluation in fields such as unsupervised information extraction, the number of TN decisions is much bigger than TP decisions. However, the number of TP decisions is more relevant than the number of TN decisions and we therefore consider that the *F-measures* are more representative than the *Rand Index* measure.

Cluster-level measures

Rather than examining all pairs of relation instances, clustering quality can be evaluated directly at the cluster level with measures such as *Purity*, or *Normalized Mutual Information* (NMI). A pre-required step for computing such measures is to assign each result cluster to

a reference cluster. The simplest strategy for performing such assignment is to choose the reference cluster that shares the largest number of relation instances with the considered result cluster (Manning et al., 2008). In this case, one reference cluster could be assigned to multiple result clusters while each result cluster is associated with its unique reference cluster (as shown in Figure 5.1). This assignment is designed for examining the precision of each result cluster, so that a global cluster-level precision can be given. The *Purity* measure adopts this assignment strategy with the definition:

$$Purity(\Omega, \mathbb{C}) = \frac{1}{N} \sum_{k} \max_{j} |w_k \cap c_j| \tag{5.7}$$

where $\Omega = \{w_1, w_2, ...w_K\}$ is the set of result clusters and $\mathbb{C} = \{c_1, c_2, ..., c_J\}$ is the set of reference clusters.

The definition of *Purity* is based on the common relation instances between each result cluster and its assigned reference cluster. It penalizes the noise inside each result cluster but it has a bias since its value tends to increase when a large number of clusters of rather small size are formed. The extreme case is that when each relation instance forms its own cluster, the *Purity* value is equal to 1.

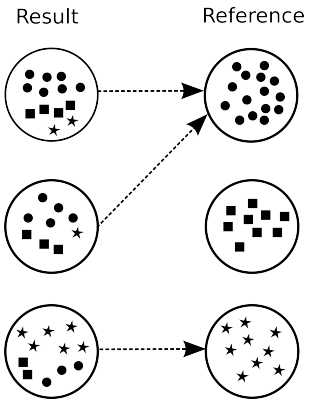


Figure 5.1: Reference assignment strategy for *purity* measure

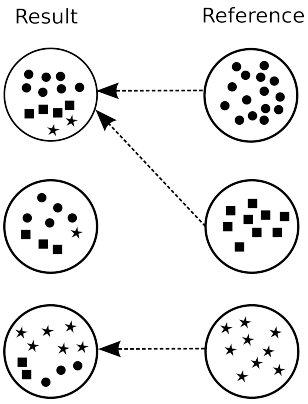


Figure 5.2: Reference assignment strategy for *inverse purity* measure

On the contrary, another class assignment strategy is to link each reference cluster to the result cluster which has the largest number of relation instances in common with it. In this case, multiple reference clusters may be assigned to the same result cluster and certain result clusters may not be linked with any reference cluster (as shown in Figure 5.2). With this assignment, the objective is to verify the recall of each reference cluster in order to give a global cluster-level recall measure. It is adopted by the *Inverse Purity* measure (Amigó et al., 2009), which is defined as:

$$Purity_{inv}(\Omega, \mathbb{C}) = \frac{1}{N} \sum_{j} \max_{k} |w_k \cap c_j| \tag{5.8}$$

Inverse Purity measure tends to favor big clusters but does not penalize mixing different category of clusters together. The extreme case is forming a single cluster including all instances, which yields a maximum of *Inverse Purity* with the value 1.

Precision and *recall* measure the performance of clustering at a level of relation instance pairing whereas *Purity* and *Inverse Purity* provide an equivalent point of view at a level of cluster of relation instances. The *Purity* and *Inverse Purity* can be respectively regarded as the cluster-level *precision* and the cluster-level *recall*.

Normalized Mutual Information (NMI) measure makes a trade-off between the number of clusters and their quality. Given result clusters Ω obtained by clustering algorithm and annotated reference clusters $\mathbb{C}$, the *Mutual Information* (MI) measure is given by:

$$MI(\Omega, \mathbb{C}) = \sum_{k} \sum_{j} P(w_k \cap c_j) \log \frac{P(w_k \cap c_j)}{P(w_k) * P(c_j)} \tag{5.9}$$

where $P(w_k)$, $P(c_j)$ and $P(w_k \cap c_j)$ are respectively the probabilities of a relation being in a result cluster w_k, in a reference cluster c_j and in the intersection of the two. The probabilities are estimated directly from the cardinalities of the clusters using a for maximum likelihood estimation (i.e. each probability corresponds to relative frequency).

NMI measure (Witten and Frank, 2005) incorporates entropy information into MI measure as a normalization factor so that the measure is in the interval [0,1], which makes this measure more interpretable then MI measure for comparisons among different experiments. The NMI measure is defined as:

$$NMI(\Omega, \mathbb{C}) = \frac{MI(\Omega, \mathbb{C})}{(H(\Omega) + H(\mathbb{C}))/2} \tag{5.10}$$

where $H(\Omega)$ and $H(\mathbb{C})$ are respectively the entropy of result clusters Ω and of reference clusters $\mathbb{C}$, calculated by:

$$H(\Omega) = -\sum_{k} P(w_k) \log P(w_k) \tag{5.11}$$

$$H(\mathbb{C}) = -\sum_{j} P(c_j) \log P(c_j) \tag{5.12}$$

5.2.3 Reference Clusters Building

A gold standard is necessary to apply external evaluation measures. We present in this section the reference we built for this objective. Generally, the reference for clustering evaluation must be carefully constructed in a way that integrates the following three considerations:

- **Quantity**: it must contain a large number of clusters with a reasonable size in order to make the evaluation representative;
- **Variety**: a certain variety of expression must exist among the relation instances in each cluster in order to take into account several ways of expressing the relation that are semantically equivalent (paraphrases) and have a richer and more realistic reference;
- **Proportionality**: each expression of a relation in a cluster must be represented in a balanced way in order to avoid potential biases: some expressions may be a lot more frequent than others but we do not want their contribution to the evaluation to be predominant, so that the capacity of the clustering to group different expressions of a relation can be evaluated with less bias.

Considering these three requirements, the reference is built in an iterative way, with human supervision for each step.

Relation querying and cluster annotation

The number of relation instances extracted from a corpus is generally very large. Hence, the construction of reference clusters of relations starts with the indexing of these relations by a search engine to facilitate the access to relation candidates. This indexing takes distinctly into account the components of a relation instance, in order to let the annotators query them specifically: the named entities ($E1$ and $E2$), the named entity types ($T1$ and $T2$) and the linguistic characterization of the relation instance (*Cmid*). The following bootstrapping procedure is then applied, relying on indexed relations:

1. **Initial query**: query the indexed relations by setting one or more fields among $T1$, $T2$, $E1$, $E2$ and $Cmid$;
2. **Result ranking**: rank resulting relation instances following the decreasing frequency of their expressions ($Cmid$ part);

3. **Cluster annotation**: choose interesting relation instances to form a new cluster or add them to existing clusters;

4. **Query update**: enlarge the set of frequent relations by updating the initial query with the characteristics of retrieved relation instances.

The first step of *initial query* produces a list of targeted relation instances. These relation instances are presented in groups of relation instances sharing the same $Cmid$ part. The second step of *result ranking* gives an order for grouped relation instances according to the size of the group. Larger groups are ranked first and exhibited at the top of the list in the user interface. In practice, this simple ranking algorithm is important and effective to make frequent relations emerge out of all other relations, so that the next step of *cluster annotation* is facilitated. The step of *query update* allows to use the retrieved (or annotated) relation instances to create new queries for finding new relation instances that share similar characterizations with the previously retrieved candidates, that can be the named entity types, the named entities or the $Cmid$ part. The steps 2, 3 and 4 can be repeated until the annotators consider that the size of the reference cluster is big enough for evaluations.

An example is given in Figure 5.3 to illustrate this iteration procedure.

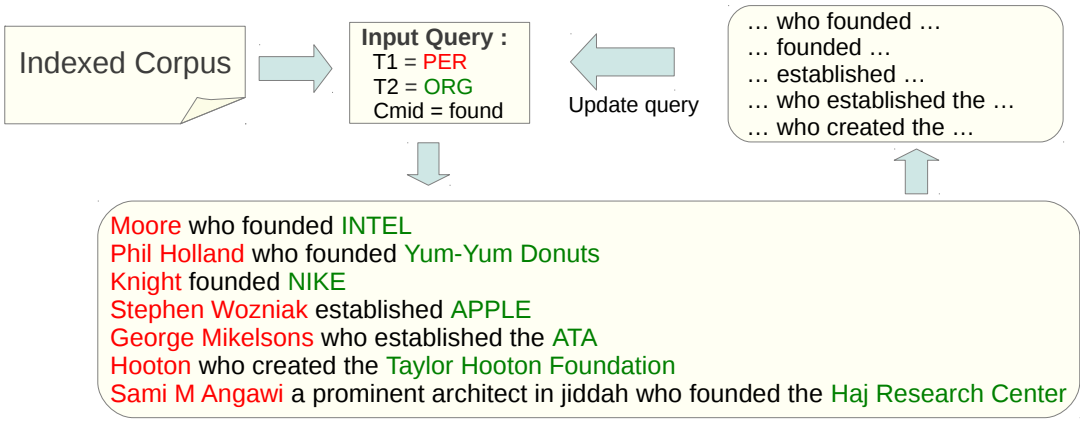


Figure 5.3: An example of bootstrapping for building reference clusters

The initial input query is given as:

"$T1$=PER, $T2$=ORG, $Cmid$=*found*"

An ensemble of relation instances in the category PER-ORG is retrieved and ranked according to the frequency of their expression ($Cmid$). These relation instances can then be added to an existing cluster or used to create a new cluster. Then, new queries can be built

for exploring all possible relations with the named entity pairs of these retrieved relation instances:

"$E1$=*Moore*, $E2$=*Intel*",
"$E1$=*Knight*, $E2$=*NIKE*", ...

The same relation tends to repeat between the same named entity pairs, with possibly different expressions. This exploration based on named entity pairs helps to find various forms of mentions for a given relation, which tackles the reference *variety* issue.

The results retrieved with these new queries are ranked once again according to the frequency of their $Cmid$ part. Different $Cmid$ parts in relations can also be chosen for building new queries for the next iteration, such as:

"$T1$=PER, $T2$=ORG, $Cmid$=*establish*" ,
"$T1$=PER, $T2$=ORG, $Cmid$=*create*" , ...

The reuse of these $Cmid$ words aims at retrieving more relation instances of the same relation type, which tackles the reference *quantity* issue for different types of expressions.

With the help of a search engine, the size of clusters can be easily enlarged, especially for high-frequency relations, so that the *quantity* issue is not practically an obstacle. However, the number of relation instances of each cluster is restricted so that evaluations will not be dominated by large clusters of too similar relations. In practice, relation instances with the same expression are limited to 30. This constraint is set for the reference *proportionality* issue.

It must be noted that all fields in the results can be used to create new queries in the bootstrapping procedure, such as

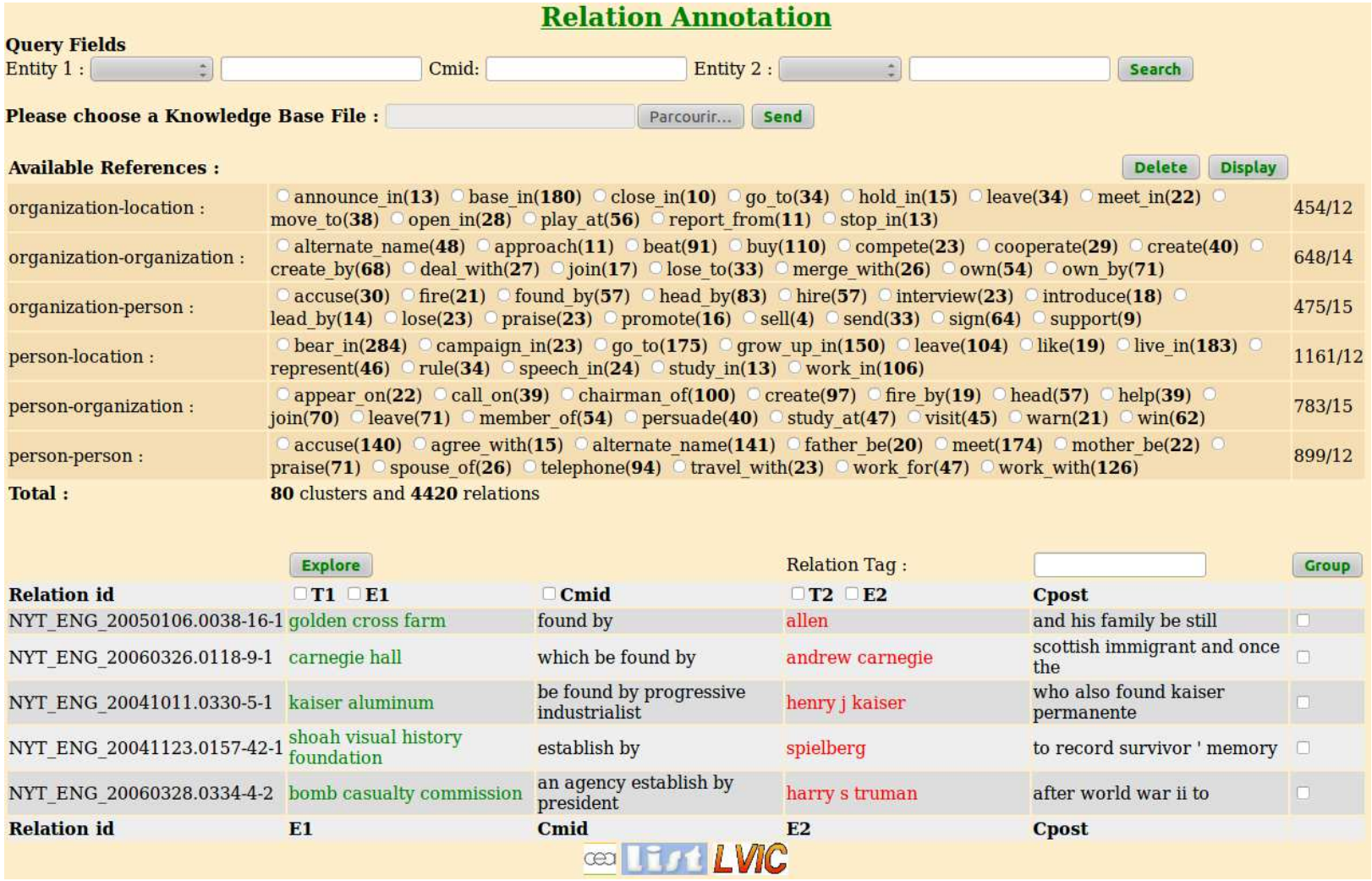
"$T1$=PER, $Cmid$=*establish*, $E2$=*Apple*",
"$T1$=PER, $T2$=ORG, $Cmid$=*who found*", ...

This allows users to discover freely the different relations indexed in the corpus.

Relation Query Tool

A Web-based relation query tool, with the interface shown in Figure 5.4, was developed for supporting this iterative procedure of reference cluster annotation. This tool integrates first of all the search engine *Lucene*[2] for querying indexed relations. Users can query relations by specifying named entity types or other fields, and then view the retrieved relation instances and group similar ones together by giving a relation label for a new cluster or adding to an existing cluster.

[2]http://lucene.apache.org



Relation Annotation

Query Fields

Entity 1 : Cmid: Entity 2 : Search

Please choose a Knowledge Base File : Parcourir... Send

Available References : Delete Display

organization-location :	announce_in(**13**) base_in(**180**) close_in(**10**) go_to(**34**) hold_in(**15**) leave(**34**) meet_in(**22**) move_to(**38**) open_in(**28**) play_at(**56**) report_from(**11**) stop_in(**13**)	454/12
organization-organization :	alternate_name(**48**) approach(**11**) beat(**91**) buy(**110**) compete(**23**) cooperate(**29**) create(**40**) create_by(**68**) deal_with(**27**) join(**17**) lose_to(**33**) merge_with(**26**) own(**54**) own_by(**71**)	648/14
organization-person :	accuse(**30**) fire(**21**) found_by(**57**) head_by(**83**) hire(**57**) interview(**23**) introduce(**18**) lead_by(**14**) lose(**23**) praise(**23**) promote(**16**) sell(**4**) send(**33**) sign(**64**) support(**9**)	475/15
person-location :	bear_in(**284**) campaign_in(**23**) go_to(**175**) grow_up_in(**150**) leave(**104**) like(**19**) live_in(**183**) represent(**46**) rule(**34**) speech_in(**24**) study_in(**13**) work_in(**106**)	1161/12
person-organization :	appear_on(**22**) call_on(**39**) chairman_of(**100**) create(**97**) fire_by(**19**) head(**57**) help(**39**) join(**70**) leave(**71**) member_of(**54**) persuade(**40**) study_at(**47**) visit(**45**) warn(**21**) win(**62**)	783/15
person-person :	accuse(**140**) agree_with(**15**) alternate_name(**141**) father_be(**20**) meet(**174**) mother_be(**22**) praise(**71**) spouse_of(**26**) telephone(**94**) travel_with(**23**) work_for(**47**) work_with(**126**)	899/12
Total :	**80** clusters and **4420** relations	

Explore Relation Tag : Group

Relation id	**T1 E1**	**Cmid**	**T2 E2**	**Cpost**	
NYT_ENG_20050106.0038-16-1	golden cross farm	found by	allen	and his family be still	
NYT_ENG_20060326.0118-9-1	carnegie hall	which be found by	andrew carnegie	scottish immigrant and once the	
NYT_ENG_20041011.0330-5-1	kaiser aluminum	be found by progressive industrialist	henry j kaiser	who also found kaiser permanente	
NYT_ENG_20041123.0157-42-1	shoah visual history foundation	establish by	spielberg	to record survivor ' memory	
NYT_ENG_20060328.0334-4-2	bomb casualty commission	an agency establish by president	harry s truman	after world war ii to	
Relation id	**E1**	**Cmid**	**E2**	**Cpost**	

Figure 5.4: Interface of relation query tool

In addition to user queries, this tool can also take as input a list of named entity couples extracted from a knowledge base. For example, the *InfoBoxes* of Wikipedia can be used to generate lists of named entity pairs for different types of relations, which serve as queries to locate relation instances in our corpus. This procedure is also a way to enlarge the variety of considered relation. The *InfoBoxes* knowledge base is based on the version provided in the *Slot Filling* task of the Knowledge Base Population track of the 2011 Text Analysis Conference[3]. Relations are mainly about the attributes for persons (*e.g. parents, children, city of birth, member of, etc.*) and organizations (*e.g. members, found by, city of headquarters, etc.*)

Annotated Clusters

With the help of our relation query tool, relation instances can be quickly found and annotated into clusters. A part of one reference cluster is shown as an example in Figure 5.5, referring to the relation *grow_up_in* of category PER-LOC.

Ms. Edwards who grow up in *Raleigh* and graduate from Princeton university ...
Bill Tosh a protestant businessman who grow up in *Londonderry* in northern Ireland be now ...
Jameelah Lewis an African-American raise in *Ohio* come seek the Judaic roots ...
Russell who be raise in *New York City* do not start his movie ...
John Kerry come from *New England*, where people don't talk about personal things like religion very easily in public ...
Pat Priest, a retire Democratic judge from his hometown of *San Antonio*, to hear the case ...
...

Figure 5.5: Reference cluster example for relation *grow_up_in* of the category PER–LOC

Our current reference is made of 80 clusters of 4,420 relations. About a dozen clusters have been constructed for each relation category with sizes varying from 4 to 280 relation instances. More precise statistics about these reference clusters can be seen on the top part of the annotation interface in Figure 5.4, the three columns of each line corresponding respectively to the relation category, annotated relations with the number of relation instances and the total number of relation instances with the number of clusters.

[3] `http://nlp.cs.qc.cuny.edu/kbp/2011`

5.2.4 The Outline of Experiments

Our experiments are evaluated within this evaluation framework. In this section, we give a synthesis of all the experiments presented in the following of this chapter, including the evaluation of the impact of the filtering procedure on relation clustering and the experiments of basic clustering, semantic clustering and topic-based relation clustering.

For preliminary experiments of basic clustering and context clustering, MCL algorithm with the binary weighting strategy was adopted because the All Pair Similarity Search (APSS) algorithm used for similarity computation is more efficient for binary vectors than weighted vectors due to its specific optimization for binary vector data (Bayardo et al., 2007) and that our early implementation of APSS was based on the computation of the binary vectors. As a result, some of the semantic clustering and topic-based relation clustering experiments are based on the results obtained by binary weighting MCL algorithm.

Impact of filtering MCL algorithm with binary weighting was applied on all the relation instances initially extracted and the remaining relation instances after the filtering procedure for comparison in order to evaluate the impact of the filtering procedure on the results of relation clustering.

Basic clustering Preliminary basic clustering was based on the binary weighting MCL algorithm, where the influence of pruning thresholds and the inflation values were analyzed. The performance of MCL algorithm using different weighting strategies (binary, tf-idf, POS) were compared as shown in Figure 5.6.

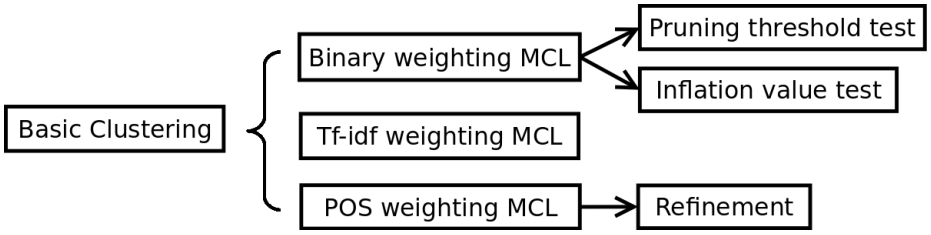


Figure 5.6: Experiments of basic clustering

Semantic clustering Preliminary semantic clustering experiments were based on the basic clustering using binary weighting for MCL algorithm. A base-line was built from this version of basic clusters using a synonym-based similarity measure between basic cluster. Different semantic similarities (WordNet-based, distributional) were then tested and compared with the base-line.

Later semantic clustering experiments were applied on the basic clusters obtained from POS weighting MCL algorithm which achieves a better performance than binary weighting MCL algorithm. Different semantic similarities were also tested for semantic clustering. In addition, the influence of verbs and nouns on semantic clustering was analyzed. The outline of these experiments in two different stages is shown in Figure 5.7.

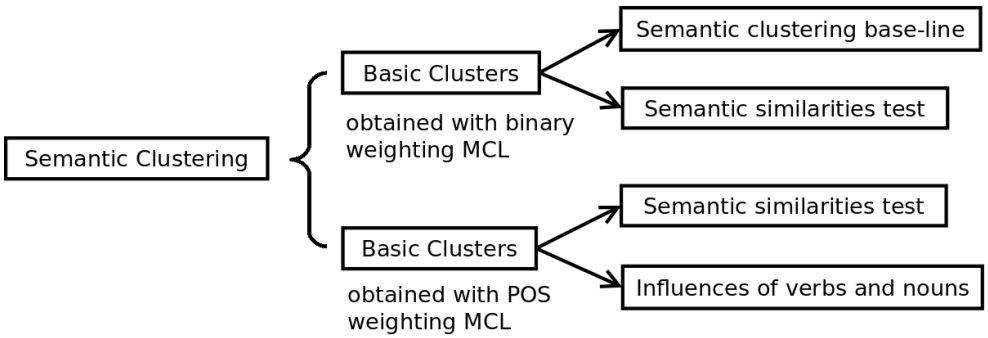


Figure 5.7: Experiments of semantic clustering

Topic-based relation clustering MCL algorithm with both binary weighting and tf-idf weighting were applied for context clustering. Preliminary context clusters were obtained by binary weighting MCL algorithm, based on which the sequential application of context clustering and relation clustering was applied (in this stage, there was only basic clustering for relation clustering). Later topic-based relation clustering includes the combination of the context clusters obtain with tf-idf weighting MCL algorithm and the best semantic clusters among all semantic clustering experiments. The Figure 5.8 shows the outline of all experiments of topic-based relation clustering.

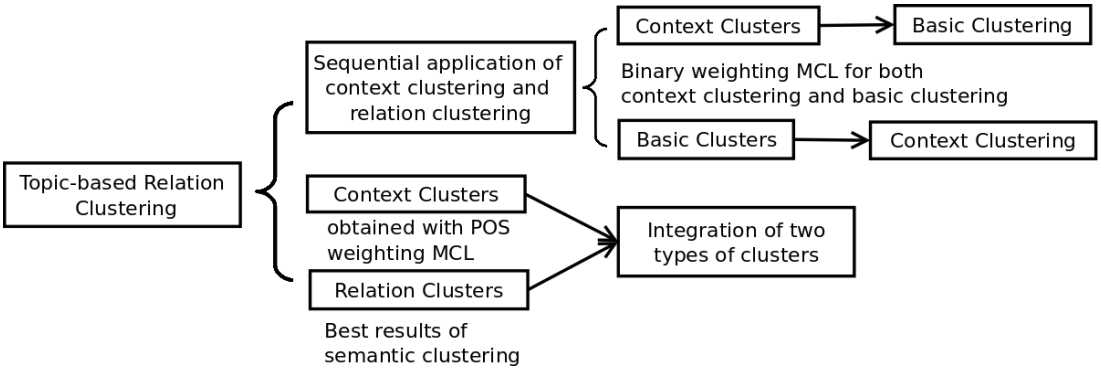


Figure 5.8: Experiments of topic-based relation clustering

5.3 The Impact of Filtering Procedure on Relation Clustering

In the relation extraction step presented in Chapter 3, a filtering procedure was applied to eliminate false relations, the performance of which was evaluated using a F-measure score on the filtering quality. Here, we present an indirect way of evaluating this filtering procedure by analyzing its impact on relation clustering. In fact, we used the same basic clustering method on all the initially extracted relation candidates (pre-filtered) and all relation instances after the filtering procedure (post-filtered) to compare the performances of basic clustering on the two different data sets. Our hypothesis is that the filtering procedure has a positive influence on the relation clustering process.

In this experiment, the similarity between relation instances is calculated with the *Cosine* measure on the $Cmid$ part and MCL algorithm with binary weighting is then applied, for both pre-filtered relation instances and post-filtered relation instances. The threshold given to APSS for similarity computation is 0.45 and the inflation value of MCL is set to 2^4. Evaluations using both internal measures and external measures are given.

5.3.1 Evaluation with Internal Measures

The internal measures *Expected Density* and *Connectivity*, presented in Section 5.2.1, are used for the evaluation. For the *Connectivity* measure, a random sample of 5,000 relation instances are selected from both pre-filtered and post-filtered relation instances and 20 neighbours (the parameter p) are taken into account. Results of these two measures are presented in Table 5.1.

Results show the positive impact of the filtering procedure. Using the same clustering method and similarity measure, clusters built from relation instances after the filtering step are generally better compared to clusters built from all relation instances before the filtering step. The two relation categories that do not follow the same tendency are, for the expected density, ORG−LOC and PER−LOC. Since both share the same entity type *location*, this observation probably indicates a special behavior of these entities. More precisely, location entities are often included in adverbial phrases, in which case there is no real relation between the location entity and the other entity. However, with the current similarity measure,

[4]The setting of these two parameters are identical in order to compare the same clustering procedure on pre-filtered corpus and post-filtered corpus. More details about the choice of these two parameters will be discussed in Section 5.4

	Expected density		*Connectivity* ($p = 20$)	
Category	**Pre-filtered**	**Post-filtered**	**Pre-filtered**	**Post-filtered**
ORG – ORG	1.06	**1.13**	5,335.7	**3,450.8**
ORG – LOC	**1.13**	1.02	4,458.7	**2,837.6**
ORG – PER	1.09	**1.17**	3,025.4	**1,532.4**
PER – ORG	1.02	**1.06**	5,638.0	**4,620.0**
PER – LOC	**1.08**	1.07	5,632.5	**4,571.3**
PER – PER	1.13	**1.15**	3,892.7	**2,569.2**

Table 5.1: Impact of the filtering procedure on clustering results: evaluation with internal measures *Expected density* and *Connectivity*

phrases with similar location adverbial phrases can be clustered together and obtain a good clustering score.

5.3.2 Evaluation with External Measures

We also evaluated the impact of the filtering procedure on the basic clustering results by comparing the results with the reference we built. A first verification of clustering results is done by computing how many relation instances from the reference are grouped by the clustering algorithm (*i.e.* are contained in a cluster of size ≥ 2). The results are provided in Table 5.2 for the clustering applied on both corpus before and after the filtering procedure.

Category	**Reference**	**Pre-filtered**	**Post-filtered**
ORG – ORG	454	307 (67.7%)	330 (72.7%)
ORG – LOC	648	485 (74.8%)	509 (78.5%)
ORG – PER	475	269 (56.6%)	286 (60.2%)
PER – ORG	1161	987 (85.0%)	998 (86.0%)
PER – LOC	783	597 (76.2%)	623 (79.6%)
PER – PER	899	586 (65.1%)	641 (71.3%)

Table 5.2: Impact of the filtering procedure on clustering results: global coverage of relation instances in reference that are in result clusters

The second column corresponds to the number of relation instances in all clusters for this relation category while the last two columns show the number of relation instances which are grouped by basic clustering (not isolated in a cluster of size 1) for pre-filtering step and post-filtering step respectively. Less relation instances are left isolated by the clustering algorithm after the application of the filtering procedure. This confirms the global

trend of the evaluation with internal measures: similar relation instances are more likely to be grouped after the filtering step because there is less noise in the set of relation instances.

The two types of result clusters obtained by the basic clustering method are then evaluated using the reference clusters with different external measures. Results of *Rand Index* and F-measures are shown in Table 5.3 and the number of different decisions used to compute these measures are shown in Table 5.4.

Category	Step	Rand Index	Precision	Recall	F_1-measure
ORG-LOC	pre-filtered	0.849	0.977	0.246	0.393
	post-filtered	0.888	0.956	0.456	**0.618**
ORG-ORG	pre-filtered	0.933	0.984	0.309	0.471
	post-filtered	0.936	0.974	0.344	0.509
ORG-PER	pre-filtered	0.914	*0.910*	0.131	0.228
	post-filtered	0.916	*0.932*	0.152	0.262
PER-LOC	pre-filtered	0.887	*0.676*	*0.409*	0.510
	post-filtered	0.899	*0.785*	*0.406*	0.535
PER-ORG	pre-filtered	0.918	0.466	0.220	0.299
	post-filtered	0.909	0.395	0.274	0.323
PER-PER	pre-filtered	0.885	0.906	0.109	0.194
	post-filtered	0.885	0.875	0.120	0.211
ALL	pre-filtered	0.981	0.708	0.282	0.403
	post-filtered	**0.982**	**0.756**	**0.312**	**0.442**

Table 5.3: Impact of the filtering procedure on clustering results: evaluation with external measures *Rand Index* and F-measures

For the evaluations including all the resulting clusters of different relation categories (the line ALL of Table 5.3), we can observe an improvement of all these four measures, in particular, with an augmentation of the number of TP decisions (Table 5.4). This amelioration of TP decisions is especially remarkable for the relation category ORG–LOC. This confirms the hypothesis that invalid relation instances have a negative influence on the clustering of relation instances.

For detailed evaluation of each relation category, we can also note a satisfying level of precision for both results before and after relation filtering, especially for categories such as ORG–ORG, ORG–LOC, ORG–PER and PER–PER. More precisely, the filtering procedure has globally a small negative impact on clustering precision but this impact is very limited for relatively high precision values. The precision is even stronger for relation categories as ORG–PER and PER–LOC, which contributes to the global amelioration of precision measure. The performance of recall measures is improved in general case, especially for the category ORG–LOC, where the recall is almost doubled. In fact, for the category ORG–LOC,

Category	Step	TP	FP	FN	TN
ORG-LOC	pre-filtered	5,029	120	15,416	82,266
	post-filtered	9,332	430	11,113	81,956
ORG-ORG	pre-filtered	6,264	100	13,982	189,839
	post-filtered	7,002	189	13,335	189,750
ORG-PER	pre-filtered	1,430	*141*	9,519	101,485
	post-filtered	1,668	*122*	9,281	101,504
PER-LOC	pre-filtered	*39,525*	*18,981*	*57,009*	*557,865*
	post-filtered	*39,197*	*10,753*	*57,337*	*566,093*
PER-ORG	pre-filtered	5,363	6,149	19,006	275,635
	post-filtered	6,667	10,192	17,702	271,592
PER-PER	pre-filtered	5,616	581	45,951	351,503
	post-filtered	6,181	883	45,386	351,201
ALL	pre-filtered	63,227	26,072	160,974	9,520,137
	post-filtered	**70,047**	**22,569**	**154,154**	**9,523,640**

Table 5.4: Impact of the filtering procedure on clustering results: difference of the number of TN, FP, FN and TN decisions

the number of TP decisions increases from 5,029 to 9,332, which means that the presence of invalid relation instances can prevent a large number of similar relation instances from being grouped together properly.

Table 5.5 presents the results of the evaluation with external measures at cluster level, in both pre-filtered and post-filtered steps for all the relation categories considered. The numbers of clusters and the average cluster sizes are also included in the table.

As shown in the first column of Table 5.5, *Purity* is higher in pre-filtered step than in post-filtered step. This may be because that *Purity* measure favors small clusters and the average cluster size for both steps are respectively 5.54 and 7.50, as shown in the last column of table. On the other hand, the improvement of *Inverse Purity* confirms the amelioration of *recall* in Table 5.3. The performance of *NMI* is improved in general as well. The entropy of result clusters ($H(\Omega)$) is also given in the table. A better entropy is observed for post-filtered clusters than for pre-filtered clusters and a lower entropy indicates less noise in clusters.

It is difficult in principle to correlate directly these cluster-level measures with F-measures for two main reasons. First, they can depend, as in the case of *Purity* or *Inverse Purity*, on the strategy chosen for assigning result clusters to reference clusters. Second, improvements of F-measure values tend to be more visible since this measure focuses on pairs of relation instances, whose number increases exponentially with the number of relation instances while the number of clusters increases more linearly with the number

Category	Step	Purity	$\text{Purity}_{\text{inv}}$	Entropy	NMI	Number	Size
ORG-LOC	pre-filtered	0.974	0.359	4.314	0.619	10,042	4.32
	post-filtered	0.974	0.463	3.989	0.649	1,714	5.71
ORG-ORG	pre-filtered	0.966	0.476	4.432	0.694	10,130	5.10
	post-filtered	0.954	0.515	4.251	0.704	1,545	6.07
ORG-PER	pre-filtered	0.958	0.257	4.981	0.640	8,901	4.81
	post-filtered	0.952	0.276	4.862	0.649	1,054	4.95
PER-LOC	pre-filtered	*0.901*	0.485	3.858	0.602	18,229	6.02
	post-filtered	*0.914*	0.493	3.833	0.622	4,386	8.63
PER-ORG	pre-filtered	0.824	0.382	4.386	*0.617*	13,341	7.28
	post-filtered	0.764	0.402	4.093	*0.599*	3,239	10.34
PER-PER	pre-filtered	0.923	0.256	5.230	*0.544*	21,695	5.15
	post-filtered	0.902	0.265	5.032	*0.543*	3,895	5.90
ALL	pre-filtered	**0.915**	0.381	6.218	0.743	82,338	5.54
	post-filtered	0.902	**0.407**	**6.046**	**0.750**	15,833	**7.50**

Table 5.5: Impact of the filtering procedure on clustering results: evaluation with external measures *Purity*, *Inverse Purity* and NMI

of relation instances. For example, F-measure for category ORG-LOC is almost doubled for the relation instances after filtering procedure, from 0.393 to 0.618, which corresponds to the NMI improvement only from 0.586 to 0.602. However, in general, the changes (increases or decreases) of cluster-level external measures such as *Purity*, *Inverse Purity* and *NMI* are respectively correlated to *Precision*, *Recall* and *F-measure*.

Nevertheless, the reference used for evaluation has been constructed with only a sample of relation instances. Therefore, on the one side, external measures such as F-measure can give a qualitative view of result clusters, checking especially the precision of clusters or pairs of relation instances grouped with the reference. On the other side, the manually built reference does not contain all possible variations of each cluster. Some relation instances are grouped together but are not taken into account in the evaluation using measures such as *recall* because they are not in the reference. Consequently, the statistics about result clusters offer a supplemental view of the performance: they are given in the last two columns of Table 5.5. The total number of clusters drops considerably since the number of relation instances becomes much smaller after the filtering procedure. However, the average size of clusters increase from 5.54 to 7.50, which indicates that a more accurate set of relation instances, the one after filtering, has the tendency of forming bigger clusters.

5.4 Experiments of Basic Clustering

5.4.1 Basic Clustering with Binary Weighting Configuration

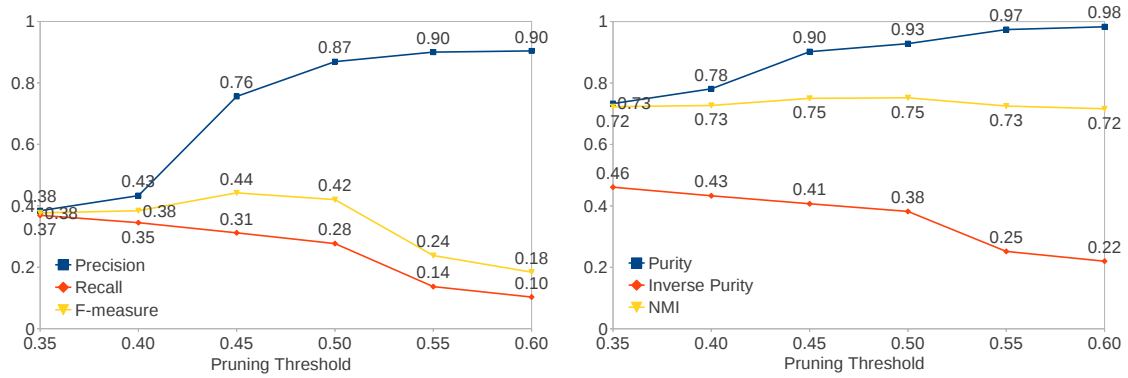
Basic clustering experiments were performed with the latest version of the MCL implementation available[5]. The first experiments are based on the binary weighting configuration, which means that every word has the same weight (w=1.0). Two important parameters to be considered are the pruning threshold and the inflation value for MCL algorithm. The influence of different values for these parameters will be discussed below.

Pruning Threshold for MCL Algorithm

The MCL algorithm makes random walks in the similarity graph so that a pruning technique make this procedure more efficient. The higher the pruning threshold is, the more efficient the MCL algorithm can be. However, if the threshold is set too high, too many edges in similarity graph would be cut, including those between similar objects. Our objective is to use this threshold to ignore those edges between the objects which are not similar at all. In our experiments, the threshold is set empirically to 0.45. This is based on observations from the Microsoft Research Paraphrase Corpus (Dolan et al., 2004) which contains an ensemble of sentence pairs. Some of these sentence pairs are paraphrases and others are not and a *Cosine* similarity computation on bag-of-word was applied on all these sentence pairs. Results show that similarity values for the pairs of paraphrase sentences are all very high while those for pairs of non-paraphrase sentences are more varied. We are more interested in the pairs of non-paraphrase sentences because it is the most dissimilar ones we want to eliminate. We observed that about 1/4 of the similarity values for non-paraphrase sentence pairs are under the threshold 0.45 and these 1/4 sentence pairs are considered as the most dissimilar ones and are supposed to be ignored. Therefore, our experiments started from this threshold value and then we also tested other thresholds around this value for comparison, the results of which are presented in Figure 5.9.

We can see clearly in these results that the *precision* measure improves when the pruning threshold gets higher, while the *recall* measure behaves in the opposite way. This observation is easy to understand since the higher the threshold is, the less noise each cluster can potentially contain, which generates a better precision. On the other side, less candidates are considered for clustering, which lowers the recall. In total, the threshold 0.45 allows to achieve the best F-measure.

[5] `http://micans.org/mcl`, the version used is mcl-12-068



(a) instance-level measures (b) cluster-level measures

Figure 5.9: Performance comparison of basic clustering using different pruning thresholds with binary weighting MCL algorithm

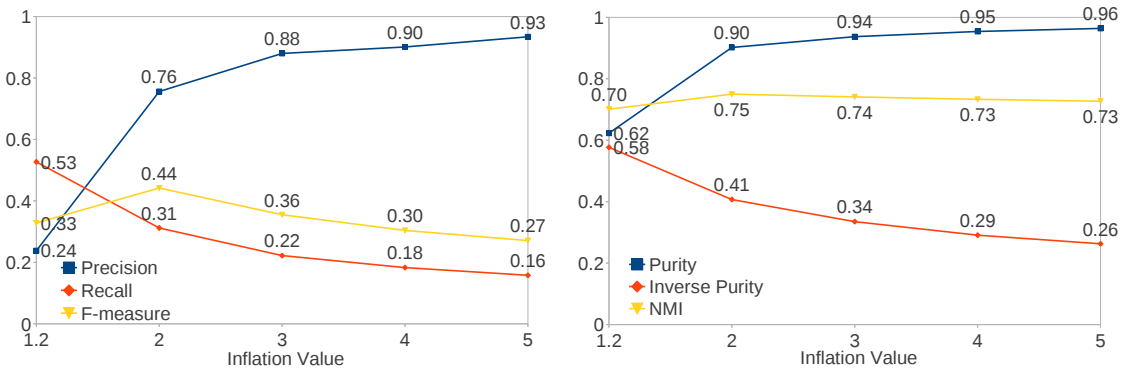
For the cluster-level measures, the *purity* and *inverse purity* measures, which reflect the cluster-level precision and recall, behaves the same way as the *precision* and *recall* respectively. A higher threshold gives a better *purity* and lower *inverse purity*. The best results of *NMI* is achieved by the thresholds 0.45 and 0.50.

This performance comparison among different thresholds confirms the empirical choice of 0.45 based on the Microsoft Paraphrase corpus as pruning threshold for the binary weighting MCL algorithm.

MCL Inflation Value

The inflation value is a single parameter controlling the granularity of clustering results of MCL algorithm and it corresponds to the power index r in the inflation step (see Formula 4.12 in Page 81). A high value tends to produce fine-grained clustering while a low value generates coarse-grained clustering results. The value is set at 2 by default and can take values in the range [1.2, 5.0]. With the fixed pruning threshold 0.45, different resulting clusters are obtained with different inflation values. The results are presented in Figure 5.10.

The default inflation value achieves the best *F-measure* and *NMI*. As we can see, the high inflation value (r=5) results in very precise clusters (*precision* and *purity*), with a sacrifice of the *recall* or *inverse purity* measure. These experiments were performed to analyze the influence of parameters on the clustering algorithm. However, the objective here is not to find the best parameters for the MCL algorithm applied on our data but to choose reasonable ones for the whole clustering procedure to a large enough number of



(a) instance-level measures (b) cluster-level measures

Figure 5.10: Performance comparison of basic clustering using different inflation values with binary weighting MCL algorithm

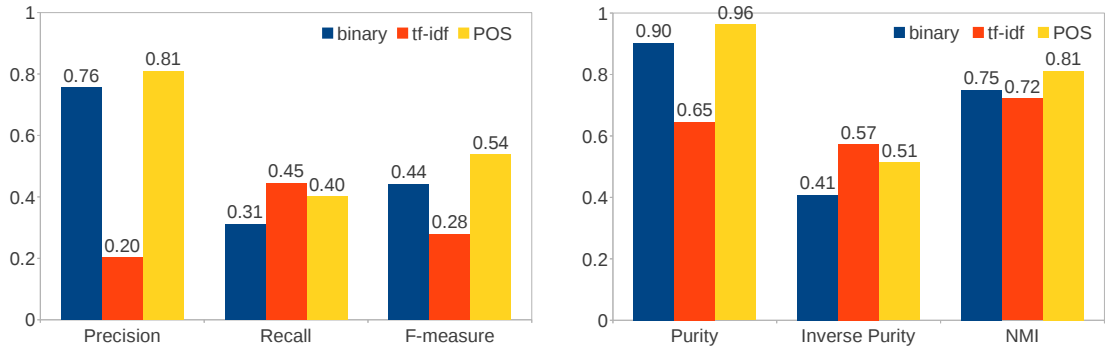
clusters in a rather precise way. Therefore, the default inflation value of 2 is used in all the following experiments using MCL algorithm.

5.4.2 Comparison of Different Weighting Strategies

The binary weighting configuration was then compared to two other weighting strategies: tf-idf weighting and POS categorization (weighting methods presented in Section 4.4). The tf-idf weighting gives an importance to each word from a general view-point, according to the frequency of this word in the relation instance and its frequency in the whole set of relation instances. POS categorization is a way of giving weights to words based on the consideration of the importance of each type of part-of-speech in relations. Different weights are given to the different types of POS, according to how important the POS type is with respect to the expression of the relation.

The MCL with tf-idf weighting takes the same pruning threshold (0.45) as the binary weighting. However, for the one with POS categorization, the threshold is augmented to 0.6 in order to take into account the looser constraints during the bag-of-words comparison. Our experiments show that this higher pruning threshold generates clusters with higher precision while keeping the same level of recall. Concerning weight setting details, different weights are adopted for different POS categories, distinguishing directly-related words, indirectly-related words, complementary words, and noise words (details in Table 4.2, Page 89)). We intuitively give them respectively a weight of 1.0, 0.75, 0.5 and 0 for these four categories and a weight of 0.5 as default. Some variations of weighting

values have been tested while no significant differences are observed as far as we keep the following principle: high weights for important categories to make them more important and low weights for non-relevant ones to make them less disturbing. The performance of these three different weighting strategies is illustrated in Figure 5.11.



(a) instance-level measures

(b) cluster-level measures

Figure 5.11: Performance comparison of MCL algorithm using different weighting strategies for similarity calculation

The MCL algorithm with the similarity calculation weighted by POS categorization outperforms the other two weighting configurations, with a better precision and a relatively satisfying recall. This is easy to comprehend since this weighting strategy emphasizes the importance of verbs, nouns, adjectives and prepositions, which are linked directly with the relation, while a smaller weight is given to words that contribute mainly to linguistic variations ("who" + verb, "the one that" + verb). This distinction enables the augmentation of pruning threshold to make more precise clusters without any big loss of recall.

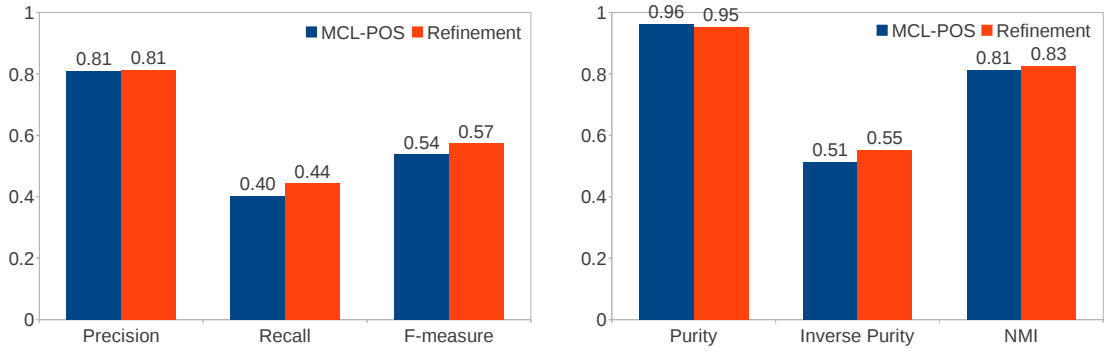
On the other hand, the tf-idf weighting does not result in satisfying performance. In fact, the term frequency is 1 inside most relation candidates. Thus td-idf weighting tends to reward words that are relatively infrequent in whole documents. However, verbs and nouns which bear the meaning of a relation are often common terms, hence have a small weight, while proper nouns that are not linked with the relation type can often obtain a relatively high weight. For example, the weight of the verb "write" is much lower than weight of the family name "Hafstrom" or the company name "Zamzow". Moreover, specific numbers can get a very high weight as well, such as the score of a basketball match "77-67", whereas this weight is set to 0 in the POS categorization weighting. As a result, the td-idf weighting perturbs the relation clustering by generating irrelevant clusters.

Although the optimization of every parameter is not the objective here, several different pruning thresholds and different weighting schemes were tested for POS categories. The current version (threshold 0.60 and weighting scheme in Table 4.2 of Page 89) gives the best performance.

5.4.3 Basic Clustering Results

The MCL algorithm with POS weighting was finally chosen for the first step of basic clustering to form basic clusters of high precision. This clustering result is complemented by a refinement to further group relation instances that are missed by MCL. This step is done by grouping basic clusters that share the same labels (the most frequent verb or noun is used as the label for each basic cluster, details in Section 4.4.2).

Performance of the MCL procedure with POS weighting on all relation candidates and its subsequent refinement are given in Figure 5.12.



(a) instance-level measures (b) cluster-level measures

Figure 5.12: Refinement of MCL results

The MCL algorithm achieves a very high precision (0.810), which indicates that the relation instances grouped inside each result cluster are rather homogeneous. The additional refinement step adds a 4 points improvement for *recall*, resulting in a 4 points of increase in *F-measure* since the high *precision* is maintained. The same improvement can be observed for *Inverse Purity* measure and *NMI* measure.

The number of clusters and the average size of all result clusters are shown in Table 5.6 for both results obtained by POS weighting MCL and those after the refinement.

It is important to note that the number of clusters after the refinement step decreases by 14.1%, from 13,648 to 11,726. The more relation instances grouped in the basic clustering

Step	Cluster number	Average size
MCL-POS	13,648	7.56
Refinement	11,726	8.80

Table 5.6: Cluster statistics of basic clustering with MCL and its refinement

step, the less pairs of basic clusters are required to be compared in the semantic clustering step, which makes the whole procedure more efficient. The average size of result clusters is raised as well. This will also improve the results of the second step of clustering procedure because the bigger basic cluster sizes are, the more redundant information can be used for semantic clustering.

After the application of basic clustering on all relation instances kept by the filtering procedure, we also performed an additional analysis using the reference clusters to verify how relation instances in the reference clusters are separated or grouped by basic clustering. This analysis shows that each reference cluster is in fact separated into several small clusters. Table 5.7 provides a more qualitative view of relation clustering results with one example for each relation category.

Relation Category	Relation	Clustering results
ORG – ORG	create	{create the}, {establish the}, {form a}, {build the}, ...
ORG – LOC	base in	{base in, a company base in}, {locate in, which be locate in}, {headquartered in}, ...
ORG – PER	found by	{found by, a group found by, be found by, which be found by}, {establish by}, ...
PER – ORG	head	{who be the head of, who head the office of}, {who head}, ...
PER – LOC	work in	{who work in, work in}, {work at, who work at}, ...
PER – PER	telephone	{call}, {who call, who call his manager}, {call president, telephone president}, ...

Table 5.7: Results of basic clustering applied on all relation instances

The second column contains one relation type annotated in our reference for each relation category and the third column gives a list of result clusters formed for all relation instances in the reference corresponding to this relation (each curly bracket pair stands for one result cluster). Each result cluster is represented by the most frequent $Cmid$ forms among various expressions existing in this cluster. Expression variations can be observed

inside each result cluster, such as for the relation *found by* in the category ORG – PER, different $Cmid$ forms can be found in the first result cluster.

A closer look at the content of clusters shows that a significant proportion of grouped relation instances share the same or very similar expressions. This is not surprising as the measure applied for evaluating the similarity between extracted relation instances is very basic and strict. However, this basic similarity measure brings a somewhat negative impact on clustering recall: it prevents some pairs of relation instances that are considered as similar in the reference from being identified. These examples also show that relation instances expressed with synonyms are not grouped in the same cluster. For instance, the relation *create* of the category ORG–ORG involves at least four different verbs for which relation instances are separated into different result clusters. One representative example for each verb variation is given hereafter in Figure 5.13.

LAPD create the *Force Investigation Division* which probe potential criminal culpability ...
University of Florida establish the *Institute of Pharmacy Entrepreneur* last year to connect young ...
Stanford University form a *Global Climate & Energy Project* to combat global warming among ...
for the Kemper Development Company, which be build a *Westin Hotel* top by 148 condos ...

Figure 5.13: Reference cluster example for relation *create* of the category ORG-ORG

The most obvious way to improve this point is to define and to use a more elaborate semantic similarity measure between extracted relation instances in order to take into account semantic phenomena such as synonymy and more complex paraphrases. Strategies for grouping *a posteriori* these basic clusters could also be considered and associated with the detection of similarity between relation instances for dealing with a wider range of expression variations. Experiments of a semantic clustering starting from basic clusters were applied with this objective.

5.5 Experiments of Semantic Clustering

The objective of the semantic clustering is to group basic clusters whose relations share the same semantic meaning. Its results depend first of all on the results of basic clustering. For preliminary experiments, the semantic clustering was applied on those basic

clusters obtained with the binary weighting MCL and these experiments will be presented in Section 5.5.1. The POS weighting MCL which achieves the best performance for basic clustering was then adopted for the last experiments of semantic clustering, which will be presented in Section 5.5.2. For all these experiments, different similarity measures at word level (WordNet based similarities and distributional similarities) or at cluster level were evaluated. Finally, some examples of semantic clusters are shown in Section 5.5.3 and an analysis of the similarity distribution on the our reference clusters is presented in Section 5.5.4 to evaluate how the relation clustering is facilitated by the two-level clustering.

5.5.1 Preliminary Experiments of Semantic Clustering

The basic clusters used for preliminary experiments of semantic clustering are obtained with binary weighting MCL. Moreover, to reduce the similarity computation between basic clusters, only verbs in each basic cluster are taken into account in these experiments. Results of semantic clustering using WordNet-based similarity, syntax-based distributional similarity and window-based distributional similarity are then compared with a synonym-based *base-line* and a manually built *best-line*.

Base-line and *Best-line*

To evaluate different semantic similarities, a *base-line* was built using a similarity measure based on only synonymous verbs in WordNet synsets. More precisely, the similarity S_w between two verbs is equal to 1 if they are in the same WordNet synset, otherwise it is equal to 0. Then the basic cluster-level similarity measure described in Formula 4.20 (in Page 92) is adopted. Therefore, the similarity $S_{C_{a,b}}$ between two basic clusters C_a and C_b is calculated using Formula 5.13 and Formula 5.14.

$$S_{W_{i,j}} = \begin{cases} 1 & \text{if } W_i \text{ and } W_j \text{ are in the same WordNet synset} \\ 0 & \text{otherwise} \end{cases} \tag{5.13}$$

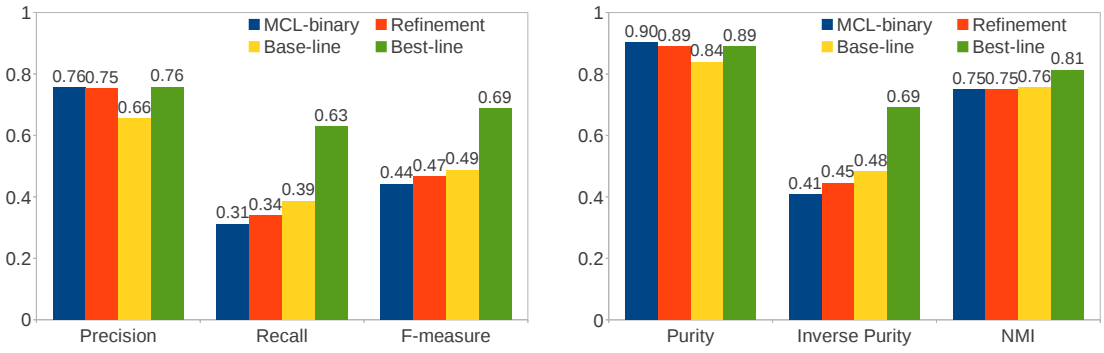
$$S_{C_{a,b}} = \frac{1}{\sum_{\substack{i\in[1,M]\\ j\in[1,N]}} f_i \cdot f_j} \sum_{\substack{i\in[1,M]\\ j\in[1,N]}} S_{W_{i,j}} \cdot f_i \cdot f_j \tag{5.14}$$

More precisely, the similarity of two basic clusters is given by dividing the number of pairs of synonymous verbs on all verb pairs between two basic clusters. As presented in Section 4.3.3 of Chapter 4, SNN clustering is adopted for semantic clustering experiments.

As we stated, the configuration of SNN clustering is highly parameterized. However, we used most of the default parameters with some variations for principal parameters especially the number of considered neighbours. The adopted configuration is: 50 for the number of considered neighbours; 0.65 for strong link percent; 0.5 for merge link percent; 0.1 for core point percent and 0.2 for noise percent. The same configuration is used for all the preliminary semantic clustering experiments of this section.

On the other hand, a *best-line* was built by manually merging basic clusters according to their matching with reference clusters. More precisely, clustering refinement was applied first by merging basic clusters sharing the same label to form new basic clusters[6]. After this refinement procedure, each basic cluster is then matched with the reference cluster which has the most relation instances in common. Then the basic clusters that share the same reference cluster are grouped to form the new *ideal semantic clusters*. The results of the *best-line* are supposed to be the best performance possible by semantic clustering, given the results of the basic clustering. To avoid too much bias during this *ideal semantic clustering*, only those basic clusters having at least two relation instances in common with one reference cluster are considered for further merging with other basic clusters.

The results for the *base-line* and *best-line* are presented in Figure 5.14, together with results of binary weighting MCL algorithm.



(a) instance-level measures (b) cluster-level measures

Figure 5.14: Performance of *base-line* and *best-line* compared to results using binary weighting MCL

[6]The refinement procedure is not applied on the basic clusters that are used for building the *base-line* because the *base-line* is one of our preliminary results and we did not repeat this experiment since it is rather time-consuming to compute all these similarities from WordNet synsets.

The synonym-based similarity measure improves *recall* (or *inverse purity*) but degrades *precision* (or *purity*). All in all, the *F-measure* is increased by 4.5 points. However, according to the *best-line*, a large margin of improvement can be expected for both *precision* and *recall*.

Evaluation of Different Semantic Similarities

Based on the basic clusters obtained after the refinement procedure on binary weight MCL results, the semantic clustering is experimented with more sophisticated similarity measures such as WordNet-based similarities and distributional similarities. Only verbs were particularly used for these similarities since relations of basic clusters are found to be mostly characterized by verbs. Concentrating on verbs makes also the similarity computation between basic clusters more efficient.

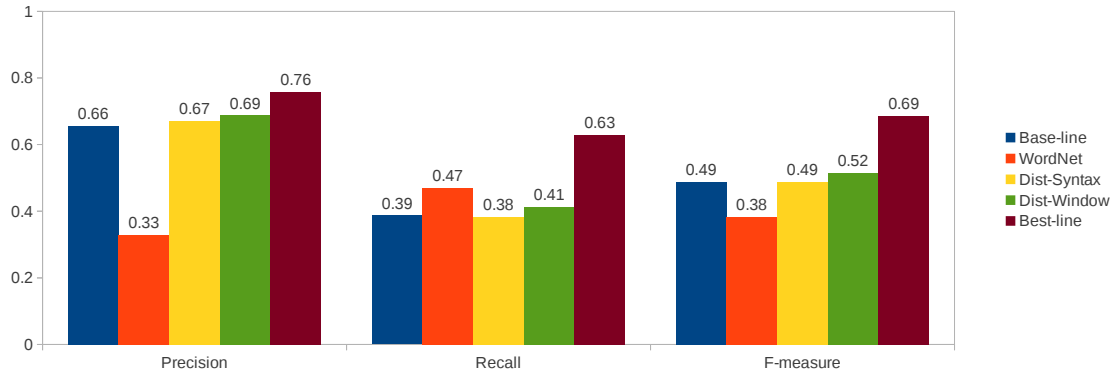
Lin similarity is chosen as the WordNet-based similarity measure between synsets of verbs, using pre-computed similarity pairs from (Pedersen et al., 2004). This similarity pair data contains about 625 million pairwise similarity values[7]. Considering that one verb may appear in several synsets, the maximum similarity of all possible synsets is taken as the similarity between words. Moreover, an empirical threshold of 0.30 is set as the minimal similarity to be considered. All verb pairs with a similarity lower than this threshold is ignored in order to emphasize the most interesting similarity pairs and to make similarity computation more efficient.

Distributional similarities are generated from a distributional thesaurus which is built from the AQUAINT-2 corpus. More precisely, for targeted words, their *context vectors* are either made of the co-occurrences in a window of size 3 (window-based similarity), which means only one nearest word considered on each side, or made of co-occurrences obtained following syntactic relations (syntax-based similarity). The *Cosine* similarities between *context vectors* are computed to produce the similarity value for corresponding targeted words. The details about the computation of these similarities are given in (Ferret, 2010). The resulting thesaurus gives for each entry its 100 most similar semantic neighbours in the descending order of their similarity with the entry. But only the first 5 ones are used for similarity calculation between basic clusters.

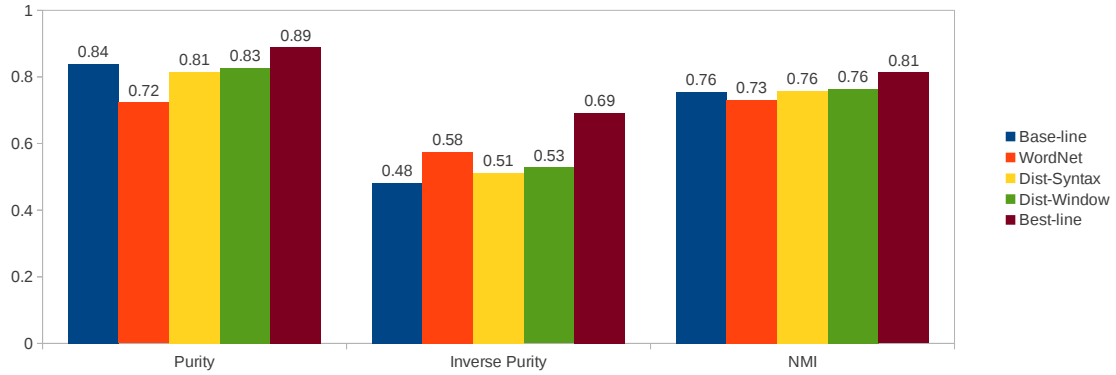
Figure 5.15 illustrates the results of different semantic similarities, along with *base-line* and *best-line* performance for comparison. As we can see, the window-based distributional similarity (Dist-Window) and syntax-based distributional similarity (Dist-Syntax) achieve

[7]Distribution of this similarity pair data is available in `http://wn-similarity.sourceforge.net`

the best performances in *F-measure* and *NMI*. They outperform the *base-line* results in *F-measure*, especially for the window-based distributional similarity. The WordNet-based *Lin* similarity (WordNet) makes the best improvement of *recall* and *inverse purity* but there is a big drop of *precision* and *purity* while distributional similarities used are more stable in these two evaluation measures.



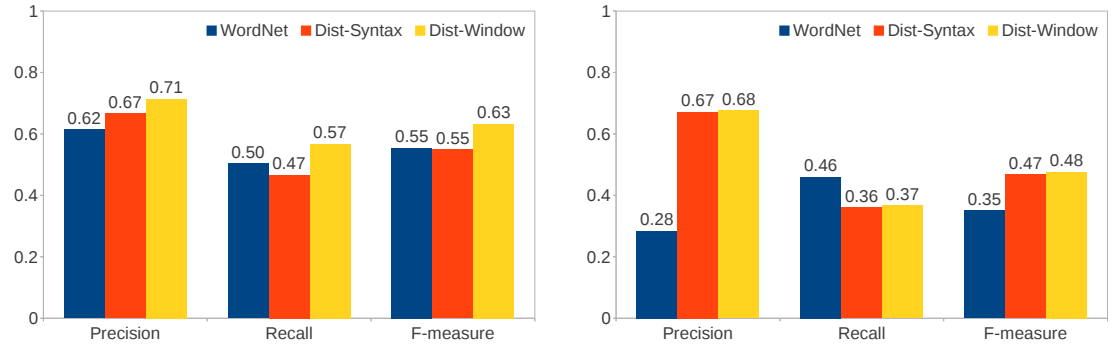
(a) instance-level measures



(b) cluster-level measures

Figure 5.15: Performance different semantic similarities for preliminary experiments of semantic clustering

A more detailed analysis of the results obtained with *Lin* similarity measure shows that its performance is very different for relation categories starting with *Organization* (ORG–LOC, ORG–ORG and ORG–PER) and those starting with *Person* (PER–LOC, PER–ORG, PER–PER). Details of these results divided into different relation categories can be seen in Figure 5.16.



(a) relation categories ORG–* (b) relation categories PER–*

Figure 5.16: Performance of different semantic similarities for semantic clustering, detailed for different relation categories

Compared to relation categories starting with PER (PER–*), the *precision* achieved by *Lin* similarity is more stable for relation categories starting with ORG (ORG–*). On the contrary, distributional similarities have a more stable performance for the different relation categories.

Table 5.8 illustrates the characteristics of the semantic clusters obtained by different semantic similarity measures, with details about the number of clusters and their average size.

	MCL-binary	**Base-line**	**WordNet**	**Dist-Syntax**	**Dist-Window**
Cluster number	15,833	10,970	6,940	9,197	9,106
Average size	7.50	10.82	17.10	12.91	13.03

Table 5.8: Characteristics of the clusters formed by semantic clustering using different similarities measures

All the semantic similarity measures used for semantic clustering, together with the base-line results obtained by WordNet synonym-based similarity are given in this table compared with basic clusters by MCL-binary. The second column in this table is the number and the average size of basic clusters obtained by binary weighting MCL. We can see that all semantic similarities reduce the number of clusters by augmenting the cluster average size. In spite of a comparable performance in the *base-line* results with other types of semantic similarities for certain evaluation measures as shown in Figure 5.15, the *base-line* results group much less basic clusters and the formed clusters are smaller than those obtained by the *Lin* similarity or distributional similarities. It also shows that the *Lin* simi-

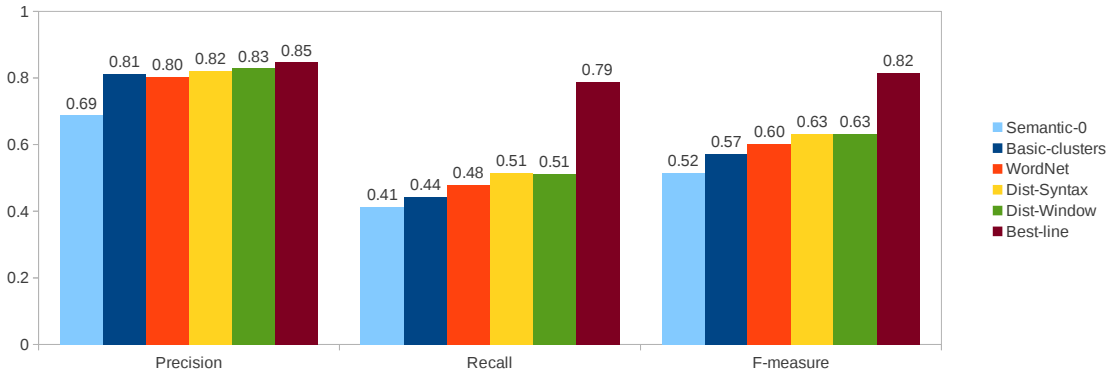
larity generates the biggest average cluster size and it groups more basic clusters than any other results, which explains its relatively high performance for the *recall* measure.

5.5.2 Semantic Clustering Using The Best Basic Clusters

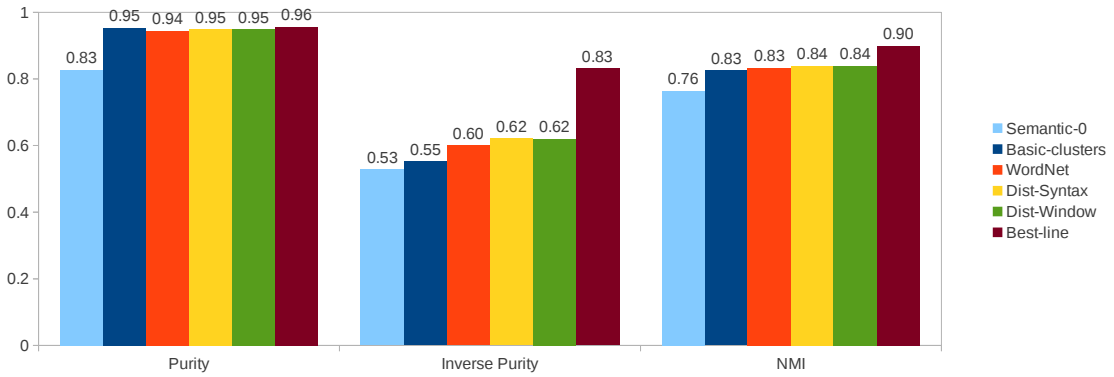
Preliminary semantic clustering experiments presented in the previous section were applied on basic clusters obtained with binary weighting MCL algorithm with a refinement procedure. In fact, we also performed semantic clustering experiments on those basic clusters generated directly by MCL algorithm without the result refinement procedure and the results of these semantic clusters are worse than the results presented in the previous section. This is easy to understand since our two-level clustering procedure takes advantage of the redundant information in basic clusters for semantic clustering so that better basic clusters produce better semantic clusters. Therefore, the amelioration of basic clusters brought by the refinement procedure leads to a better performance for the semantic clustering. Consequently, more researches have been carried out to improve basic clustering results such as the experiments of different weighting strategies presented in Section 5.4. The best basic clusters are obtained by POS weighting MCL algorithm with the refinement procedure (as shown in Figure 5.11 and Figure 5.12) so that they are finally used for the semantic clustering experiments presented in this section.

The different similarity measures used in the previous section were applied for this semantic clustering based on the best basic clusters, still focusing first on verbs for the similarity computation. Most of the parameters are the same as the ones used for preliminary semantic clustering experiments. However, the number of considered neighbours are set to 20 rather 50, since there are less neighbours which are potentially synonymous relations when bigger basic clusters are formed by the POS weighting MCL than the binary weighting MCL. We also observed in our experiments that the lower number produces more precise cluster without significant loss of recall so that the overall performance is improved. This parameter configuration is used for all these semantic clustering experiments in this section. Details of these results using different semantic similarity measures are shown in Figure 5.17.

The first columns ("Semantic-0") are the best results obtained in preliminary semantic clustering experiments of the previous section, for which the window-based distributional similarity is applied. The first thing to be noted is that all these new results even including results of basic clusters, perform better in *F-measure* or *NMI* measure compared to results of "Semantic-0". In the preliminary experiments, semantic clustering achieves an amelioration of recall measure but always with a loss of precision compared to basic clusters.



(a) instance-level measures



(b) cluster-level measures

Figure 5.17: Performance of semantic clustering using different semantic similarities, based on the best basic clusters

Here, compared to the results of basic clusters used for semantic clustering, all semantic similarity measures lead to equivalent ("WordNet") or even better ("Dist-Syntax" and "Dist-Window") *precision* performances.

The improvement of the quality of basic clusters has multiple effects. First, the *precision* of basic clusters reaches 0.812 while the earlier value of *precision* was only 0.752, which means that less noise is present among the information in each basic cluster. Moreover, the *recall* of basic clusters is improved from 0.339 to 0.442. This amelioration does not only save time for similarity computation but also makes the semantic clustering concentrate more on synonymy or paraphrase.

Among all these three different semantic similarities, distributional similarities perform generally better than WordNet based similarity, in both *F-measure* and *NMI*. In fact, syn-

onymous words included in a dictionary, such as the ones in WordNet synsets, are often too general while the corpus-based similarity measures are more directly related to the similarity between relation instances in our case, especially when the same corpus is used for distributional similarity calculation. This can explain why distributional similarity measures achieves a better *precision* or *purity*. In addition, corpus-based similarity measures go beyond the limit of synonyms since they are based on the co-occurrence vectors of words. Moreover, distributional similarities also have advantage of making the clustering methods more portable to different languages since corpus based similarities are much easier to build than experts-based resources such as WordNet.

Experiments Involving Different Categories of Words

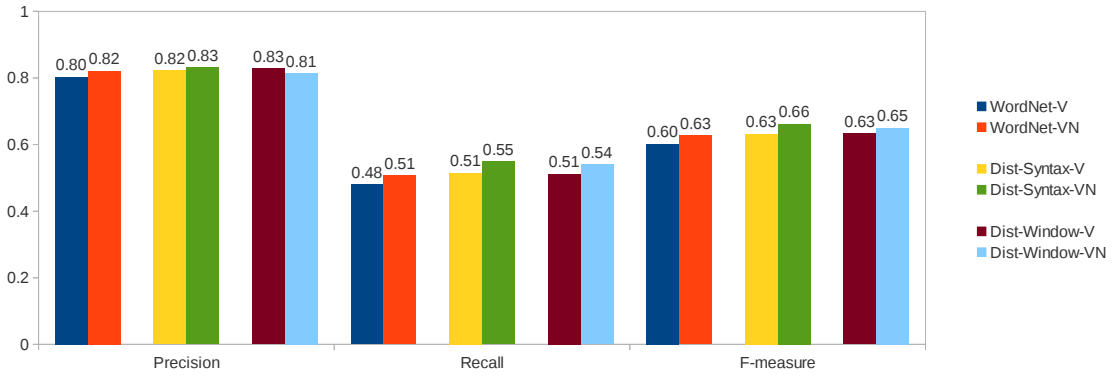
As stated before, observations show that relation instances are mainly characterized by verbs. Given a basic cluster with relation instances characterized by verbs, the number of different verbs is often very limited whereas the nouns can be very diverse, which would add a significant computation cost if all these nouns were taken into account for comparison as well. This is one reason why our experiments were first limited to verbs for similarity calculation between basic clusters.

Nevertheless, different categories of words may have different contributions to the semantic clustering since there also exists basic clusters characterized by nouns and moreover, even in those basic clusters characterized by verbs, nouns can bring sometimes important information. Consequently, a second set of experiments have been conducted including also nouns. We adopted the WordNet-Based *Wu-Palmer* (*Wup*) similarity between nouns implemented by the NLTK package[8], since we observed that *Wup* similarity is more relevant for nouns in practice. The same corpus and configuration were used to generate distributional similarities between nouns as for verbs. Therefore, besides the comparisons of verbs, nouns in one basic cluster were also compared to all the nouns in another basic cluster during each basic cluster comparison for similarity computation.

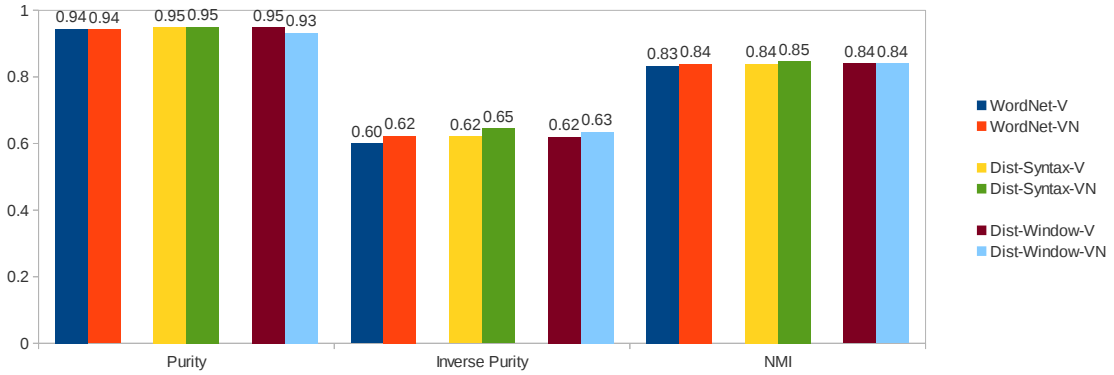
A comparison of semantic clustering with only verbs and with both verbs and nouns is illustrated by Figure 5.18.

In this Figure, every group of two bars compares one measure for one kind of semantic similarity, with the left one obtained using only verbs and the right one obtained using both verbs and nouns. In all cases, the addition of nouns helps to find more similar basic cluster pairs so that the *recall* measure and *inverse purity* measure are increased. Since the performance of *precision* measures are only slightly decreased ("Dist-Window") or

[8]Natural Language Toolkit `nltk.org`



(a) instance-level measures



(b) cluster-level measures

Figure 5.18: Evaluation of the contribution of nouns for semantic clustering

even improved ("WordNet", "Dist-Syntax"), the *F-measures* are generally improved by the employment of nouns.

The characteristics of semantic clusters are shown in Table 5.9, including both the versions with only verb and with both verbs and nouns for the three different semantic similarities.

	Basic clusters	**WordNet**		**Dist-Syntax**		**Dist-Window**	
		V	**V+N**	**V**	**V+N**	**V**	**V+N**
Clusters number	11,726	10,169	9,403	10,608	10,116	10,517	10,161
Average size	8.80	10.15	10.98	9,73	10,20	9.82	10.16

Table 5.9: The contribution of nouns in terms of cluster characteristics

Compared with the basic clusters, the same trend can be observed: a decrease of the number of clusters and an increase of the cluster average size. The last columns show the results with the three different semantic similarities, involving only verbs (V) or both verbs and nouns (V+N). As for *F-measure*, the use of nouns leads to an improvement of clusters in terms of their characteristics for all semantic similarities. This confirms the utility of nouns for semantic clustering.

Alternatively, adjectives were also tested as an additional word category for distributional similarities (verbs + nouns + adjectives), which generate almost the same clusters as the ones obtained with only verbs and nouns, in terms of both evaluation measures and cluster characteristics. This probably indicates that the adjectives appear much less frequently than verbs and nouns in the $Cmid$ part of relation instances.

We also performed semantic clustering experiments using cross category similarities, which means similarity between one verb and one noun according to their *context vectors*. Both syntax-based and window-based distributional similarities were tested but no obvious improvement was observed when using this cross category similarity.

5.5.3 Semantic Clustering Results

Table 5.10 gives a sample of semantic clusters formed using the window-based distributional similarity. Each row in table is one example of semantic cluster for the corresponding relation category, and each word in the second column stands for its basic cluster.

Relation category	**Semantic clusters**
ORG – ORG	purchase, buy, acquire, trade, own, be purchased by
ORG – LOC	start in, inaugurate service to, open in, initiate flights to
ORG – PER	sign, hire, employ, interview, rehire, receive, affiliate
PER – ORG	take over, take control of
PER – LOC	grab gold in, win the race at, reign
PER – PER	win over, defeat, beat, oust, topple, defend

Table 5.10: Examples of semantic clustering results

It is clear from these results that various words with synonymous meanings are grouped together, including synonymous verbs such as *purchase*, *buy* and *acquire*, or synonymous phrases such as *take over* and *take control of*. However, the actual semantic clustering algorithm does not treat the difference between passive and active tenses. Therefore, certain clustering errors still exist for relations having the same named entity type for both

arguments $E1$ and $E2$, for example *purchase* and *be purchased by* for the category ORG – ORG.

5.5.4 The Effects of Multi-Level Clustering

As discussed before, sophisticated semantic similarities are much more time-consuming to compute compared with simple *cosine* similarities. The total number of relation instances reaches up to 165,708 while the volume of basic clusters are only 11,726 (in Table 5.6). Hence, a first advantage of the multi-level clustering is that it reduces the computational cost of the application of semantic similarities on a large corpus.

On the other hand, another question that emerges is whether this multi-level clustering helps to ameliorate the semantic clustering results. The hypothesis is that repeated information in basic clusters help to locate interesting elements for semantic clustering. Comparison experiments were not done by applying sophisticated semantic similarity measures directly on all selected relation instances since it is almost unfeasible considering the corpus size. However, to verify this hypothesis, the annotated reference clusters were used to compare the distribution of similarities between relation instances and the distribution of similarities between basic clusters. The assumption is that the similarity distribution of basic clusters has a tendency of giving a better separation of data than the similarity distribution of relation instances.

In the first step, we calculated all the similarities between two relation instances in the same reference cluster (intra-cluster distribution D_{intra}) and the similarities between two instances in different reference clusters (inter-cluster distribution D_{inter}). This gives two distributions, D_{intra} and D_{inter}, which are expected to be well separated in the ideal case, with a high mean of similarity values for D_{intra} and low mean of similarity values for D_{inter}.

In the second step, we separated each reference cluster into basic clusters according to the basic clustering results. Since the basic clustering method tends to form small but precise clusters, each reference cluster is split into several small clusters. Then, we examined all similarities between two basic clusters in the same reference cluster and those in different reference clusters, which form a new intra-cluster distribution D'_{intra} and a new inter-cluster distribution D'_{inter}.

Figure 5.19 presents an example to demonstrate these similarity distributions, with similarity between relation instances (D) on the top and similarity between basic clusters (D') below. The example is based on the basic clusters formed with POS weighting MCL algo-

rithm with the refinement procedure and the syntax-based distributional similarity is taken for both similarity computation between relation instances and between basic clusters.

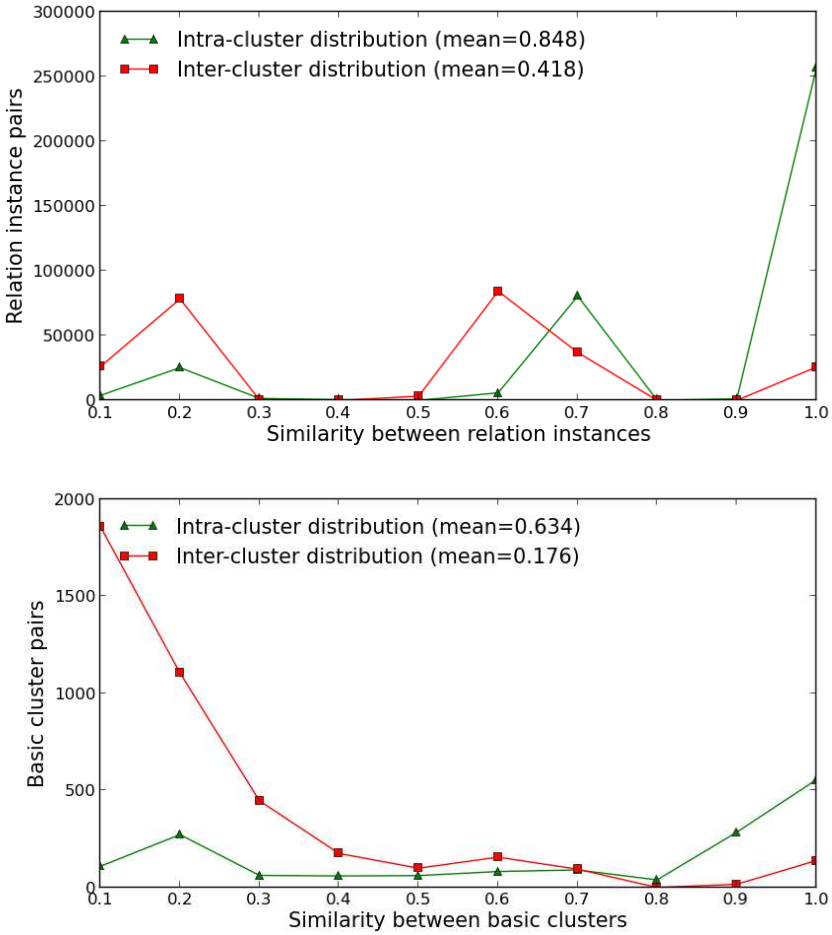


Figure 5.19: Distribution of similarities between relation instances (D) and between basic clusters (D')

It is clear to see from these figures that the semantic clustering applied on basic clusters has a better tendency of forming good clusters than applied directly on relation instances. Indeed, intra distribution and inter distribution of basic clusters are better separated than those of relation instances and the similarity between basic clusters from different references has a rather low mean value. The same phenomena of the difference between intra distribution and inter distribution is observed with all types of semantic similarities presented. This confirms the prior assumption that redundant information in basic clusters can be used to get rid of the noise brought by irrelevant words. We can also see from these figures the number of relation instance pairs and the number of basic cluster pairs in the reference: the former one is two orders of magnitude larger than the last one so that 100

times more similarity computation is required by the relation instance pair comparison than the basic cluster pair comparison.

5.6 Experiments of Topic-based Relation Clustering

Each relation instance is also characterized by a topic *context*, built from the thematic segment in which the relation instance is extracted. This topic context adds a more global perspective to the local information from the source sentence of the relation instance. The clustering of these thematic segments, called *context clustering*, helps categorize relation instances into different themes since each relation instance corresponds to one thematic segment. Topic-based relation clustering investigates more precisely how context clustering can be integrated with relation clustering. Experiments of context clustering is first presented in Section 5.6.1. In preliminary experiments of topic-based relation clustering, relation clustering and context clustering are executed sequentially, with one clustering step based on the results of the other, which will be presented in Section 5.6.2. At last, we present in Section 5.6.3 our experiments on the merging of relation clusters and context clusters built in parallel to make more precise relation clusters.

5.6.1 Context Clustering

The objective of context clustering is to group thematic segments so that each cluster of thematic segments represents a specific theme (see details in Section 4.6). Each thematic segment is represented as a bag-of-words containing all the content words in this segment. We adopted the *Cosine* measure for calculating the similarity values between different thematic segments. As for basic clustering, the All Pair Similarity Search (APSS) algorithm was used for the computation of the pairwise similarity between thematic segments. Different pruning thresholds (0.15, 0.20, 0.25, etc) and different weighting strategies (binary weighting and tf-idf weighting) were experimented in this framework. The MCL algorithm was then applied to the similarity matrix obtained by APSS to build context clusters.

Table 5.11 illustrates some context clusters obtained by applying the context clustering on the whole corpus with the tf-idf weighting and the pruning threshold 0.15. These context clusters were ranked according to the decreasing order of their size and the ones shown in the table are the largest ones. The theme that underlies each context cluster is represented by a set of characteristic words, as the contexts of relation instances. This set of characteristic words comes from the fusion of the representation of all the thematic segments in the

context cluster and the most frequent words are shown in the table. Each row in the table stands for one context cluster with the list of its characteristic words.

It has to be noted that the words with a too general sense, such as *do, be, have*, are removed from this table to make it easier to notice more meaningful words [9]. For every context cluster, the characteristic words are presented in the decreasing order of their frequency in the cluster. Words that are the most representative of the theme of each cluster are tagged manually (in bold). Moreover, Table 5.11 shows that these clusters refer quite straightforwardly to well-known topics such as *war*, *sport*, *presidential campaign*, *natural disaster*, *energy*, *economy*, *education*, *religion*, etc.

In these experiments, we did not evaluate directly the performance of context clusters obtained with different weighting strategies and different pruning thresholds. We evaluated context clustering indirectly by its influence on relation clusters when combining context clustering and relation clustering.

5.6.2 Sequential Application of Context Clustering and Relation Clustering

In these preliminary experiments, relation clustering was performed by the basic clustering with the binary weighting MCL algorithm without a second step of semantic clustering and context clustering was also based on the binary weighting MCL algorithm. The context clustering and the relation clustering were applied sequentially, one after another. When context clustering is applied first and relation clustering takes place inside each context cluster (option 1), more precise relation clusters are expected, following the assumption that relation instances of the same theme are more likely to be similar. Alternatively, if relation clustering is applied first, context clustering is applied to divide relation clusters (option 2) with the purpose of distinguishing different semantic meanings inside relation clusters according to the topic of relation instances. Both options of these two sequential clustering steps were carried out using different pruning thresholds. Their results are given in Table 5.12.

The results of these combinations are compared to the initial performance of relation clustering obtained with the binary weighting MCL algorithm ("MCL-binary"). The first thing we can notice in the table is the dramatic drop of the *recall* measure for both orders of sequential application. Whatever this order, the second clustering step divides the clusters of the first step into smaller pieces, which naturally tends to make the *recall* measure

[9] These words own a light weighting during the clustering procedure as well since they have a small idf value.

Theme	Characteristic Words
1	**iraq american** official baghdad sunni force **military** kill bush government police attack insurgent election shiite **war troops** soldier saddam
2	sox **game** yankee red **team season** run series play hit boston win pitch angel start world home come **league** manager **player**
3	bush kerry **president** state vote **election campaign** republican john percent iraq democrat war cheney debate **american** win presidential
4	orleans **hurricane** Katrina **storm** state official home federal bush louisiana house coast area water resident emergency **disaster damage**
5	**court** roberts **senate** bush **justice** president republican **case** democrat abortion right nominee conservative **law** committee issue decision **judicial**
6	**palestinian israel** gaza sharon **hamas** bank minister state election official settlement prime government **security** authority **peace** jerusalem
7	**oil** price **energy** company percent **gas** state gasoline production bill saudi barrels market drilling government bush crude world increase
8	**game** italy medal win **olympic** team world gold **olympics** turin cohen **sport** skate american kwan woman winter
9	**china** chinese **japan trade company** country percent japanese **american** government world taiwan beijing currency market dollar **export economy**
10	**tax** state cut percent house **income** bill bush pay budschool property increase senate **revenue** republican federal business plan company government money
11	**airline** delta bankruptcy **flight pilot** carrier cut percent **airway** fare company plan pay air industry union price fuel **airport** passenger
12	**school** student state **education** college teacher percent high **university** test district program public child class work official parent math time system
13	percent price sales rise **market** index report job point **increase** company home fell **growth** stock average gain rate **economy** economist consumer
14	**drug medicare fda** plan company **health** agency benefit prescription program patient percent administration bush pay federal government coverage
15	**pope church catholic** john paul vatican cardinal world St bishop priest benedict rome peter state ratzinger city

Table 5.11: Some examples of context clusters with their characteristic words, obtained by the MCL algorithm and a tf-idf weighting on segments' words

	Threshold	Rand Index	Precision	Recall	F_1-measure
MCL-binary	0.45	0.982	0.756	0.312	0.442
Option 1	0.15	0.977	0.754	0.014	0.028
	0.20	0.977	0.845	0.020	0.039
	0.25	0.977	0.849	0.015	0.029
Option 2	0.15	0.977	0.879	0.012	0.025
	0.20	0.977	0.957	0.006	0.013
	0.25	0.977	0.976	0.001	0.003

Table 5.12: Results of applying relation clustering and context clustering sequentially in different orders

decrease since relation instances that are part of the same cluster in the reference are more likely to be separated. Furthermore, the very significant drop of the recall measure shows that the sequential application of the two clustering steps generates too many clusters with a very small size.

When clustering is first applied for each relation category to form context clusters, on which relation clustering with binary weighting MCL algorithm is done subsequently (option 1), the *recall* measure is shown to be less sensitive to the application of these two sequential clustering algorithms. Nevertheless, the *recall* performance is still far from being usable in practice.

When context clustering is applied on relation clusters (option 2), the correlation between the *precision* measure and the pruning threshold for context clustering appears clearly: the higher the threshold is, the more precise the result clusters are. However, this high precision is not practically useful due to overly low *recall* (e.g. 0.001 for the threshold 0.25).

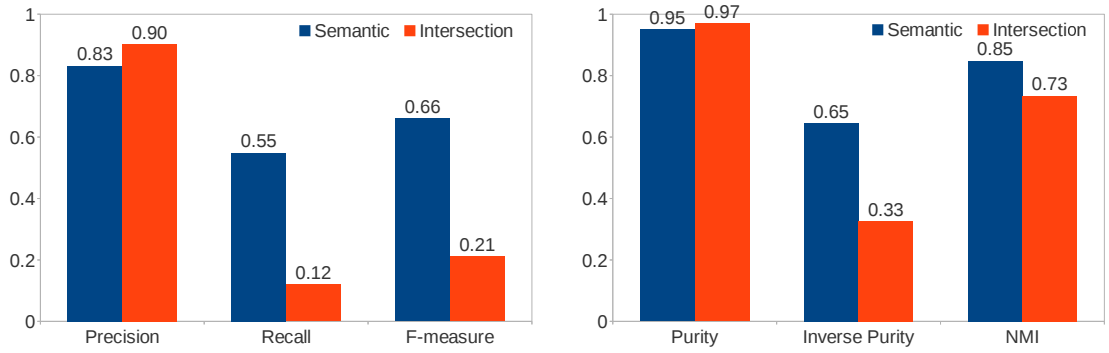
One possible reason of the low performance of the combination of context clustering and relation clustering could be the quality of context clusters. In option 2, context clustering is applied to each relation cluster and the number of thematic segments in each relation cluster is often too low to generate meaningful context clusters. In option 1, context clustering is applied for each relation category, which leads to better context clusters because of a larger number of thematic segments. This may explain why the drop of the *recall* measure is less important in this case than in option 2, especially when high values of pruning threshold are used for context clustering.

5.6.3 Integration of Context Clusters and Relation Clusters

The sequential application of two types of clustering degrades dramatically the *recall* measure. Therefore, we chose to apply context clustering and relation clustering independently and then, to integrate the resulting context clusters and relation clusters. A context cluster may contain several semantic relation types whereas a relation cluster may refer to several themes. We focused on the intersection between context clusters and relation clusters, especially by considering how relation instances in a relation cluster are grouped according to context clusters.

For these experiments, we adopted one of our best relation clustering methods, with the use of POS weighting scheme for basic clustering and the syntax-based distributional similarity for semantic clustering. For each relation cluster, if any two or more relation instances are identified as belonging to the same specific theme (grouped in one context cluster), they are separated from their initial relation cluster to form a new relation cluster. The relation instances that do not share a theme with any other relation instance in the relation cluster remain in their original cluster (as shown in Figure 4.9 in Page 98).

Results of the intersection between the two types of clusters are compared with our reference clusters to evaluate the performance of these new relation clusters. Their respective performance can be seen in Table 5.20.



(a) instance-level measures (b) cluster-level measures

Figure 5.20: Intersection of context clustering results with relation clusters

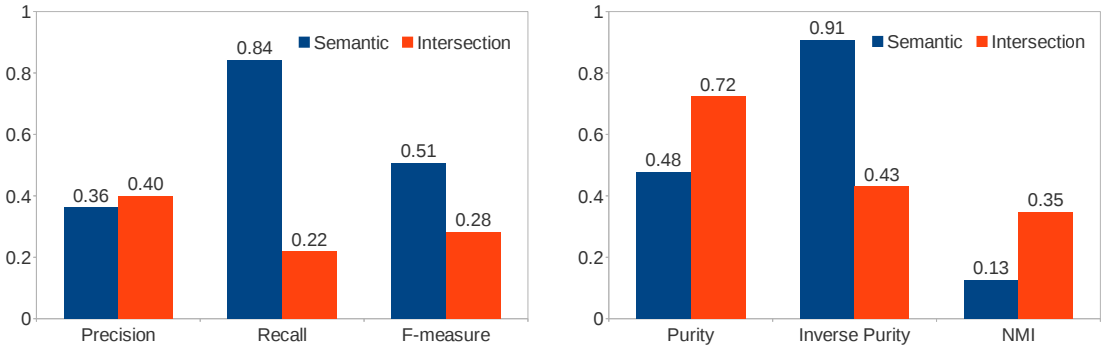
In comparison with semantic clustering ("Semantic"), the integration of context clusters and relation clusters ("Intersection") improves the *precision* measure from 0.831 to 0.902. This improvement comes unsurprisingly with a fall of the *recall* measure which, in this case, is much more satisfying than the *recall* performance achieved by the sequen-

tial application of two kinds of clustering presented in the previous section. Although the *F-measure* decreases, the primary objective of integrating context clustering and relation clustering is not necessarily to augment the overall performance of relation clusters but to use thematic information to help contextualizing more precisely relation clusters.

Evaluations with a Specific Reference

As stated at the beginning of this chapter, our reference of relation clusters was built efficiently with a Web-based tool in an interactive way. Using the same tool, we built a small specific reference which contains an ensemble of relation instances of the same semantic relation but refers to several relation clusters according the context clusters. Thus, this specific reference can be used to evaluate the impact of the context clusters when they are integrated with relation clustering results. All in all, we annotated 65 relation instances for the relation "lead_by" of the category ORG-PER. These relation instances were then separated manually into three subgroups according to the theme of the sentence in which this relation instance was found. At last, three reference clusters were built for the relation "lead_by", referring to three distinct themes: politics (30 instances), economics (21 instances) and sports (14 instances).

The evaluation on this theme-specific reference is illustrated in Figure 5.21.



(a) instance-level measures (b) cluster-level measures

Figure 5.21: Intersection of context clustering results with relation clusters, for the relation "lead_by" in different topics

We can observe the same phenomena as for the evaluation performed in the previous section: an improvement of *precision* and a drop of *recall*. However, the *purity* measure, which corresponds to a the cluster-level precision, is significantly improved (from 0.477

to 0.723), similarly to the *NMI* measure (from 0.127 to 0.348). The drop of the *recall* measure suggests that context clusters are too small, which means too many relation instances are pulled out to create new relation clusters. We then investigated further how relation instances in each reference cluster of this specific reference were divided into different themes according to the results of context clustering. Table 5.13 gives some context clusters obtained inside each reference context cluster with their characteristic words.

These context clusters are listed in the decreasing order of the number of relation instances in the reference that are in this context cluster. It appears clearly that each theme is divided into many sub-themes. For example, there are many context clusters formed for the relation instances manually linked to the theme *Sports*, which may refer to different sports including baseball, basketball, boxing, etc. Even if these themes contain common words such as *game*, *play*, *season*, etc, they are not clustered together as one theme. One factor of influence is the special names of teams or players, such as *Sox* and *Yankee* for baseball, *Lakers* and *Bryant* for basketball, *Ruiz* and *Toney* for boxing, and so on. Additionally, different sports involve different verbs and actions: for example, *hit* and *pitch* for baseball, *shot* for football and basketball, *fight* for boxing. On one hand, this thematic information is interesting since different kinds of sports can be separated. On the other hand, larger context clusters, which would group for instance different kinds of sports in the same cluster, could help for preserving a high level of *precision* while alleviating the *recall* drop problem.

5.7 Conclusions and Perspectives

In this chapter, the problem of the clustering evaluation for unsupervised information extraction was first tackled. Two complementary approaches were more particularly proposed to address it: a large-scale evaluation based on internal clustering evaluation measures that characterize to which extent clusters are representative of similarities between relations; a more restricted but deeper evaluation based on the *a priori* building of reference clusters and the use of external clustering evaluation measures. A methodology for building the reference clusters for a given corpus and an annotation tool for supporting this methodology were proposed by integrating a search engine and a simple ranking process.

The impact of the filtering step on relation clustering was first evaluated with both internal and external measures. Experiments showed that false relation instances removed by our filtering procedure have a negative influence on the relation clustering procedure. The evaluation of our various experiments concerning basic clustering and semantic clustering

Theme	Characteristic Words
Politics	
1	**iraq american** official baghdad sunni force **military** kill bush government police attack insurgent election shiite **war troops** soldier saddam
3	**oil** price **energy** company percent **gas** state gasoline production bill saudi barrels market drilling government bush crude world increase
2	**palestinian israel** gaza sharon **hamas** bank minister state election official settlement prime government **security** authority **peace** jerusalem
4	intelligence Bolton bush **house senate** state official agency **president committee** republican bill powell United national democrat commission
5	**security social** bush account **benefit** president tax **retirement** private house republican worker democrat congress investment government
Economics	
1	share **company** quarter oracle **earnings revenue** analyst report rise sales **stock** profit business executive increase price chief income market
2	**china** chinese **japan trade company** country percent japanese **american** government world taiwan beijing currency market dollar **export economy**
3	cell cancer **research human disease patient** study university breast researcher **treatment** drug bush Hwang life **medical health**
4	gm **company** delphi **car** motor **vehicle** union general **plant** auto health market share job executive benefit sales analyst bankruptcy automaker
5	morgan stanley **firm** purcell executive mack chief **investment** company board business **bank** former street deal director banker
Sports	
1	UCLA bruin **game play team** season point win coach lead come Olson score Howland Farmar player half run start shot goal end pass lose
2	sox **game** yankee red **team season** run series play hit boston win pitch angel start world home come **league** manager **player**
3	Bryant Lakers **game play** point **season** team O'Neal Odom jackson Kobe player coach quarter shot **NBA** Tomjanovich win center lead score
4	**game** louis lightning **team** cardinal **play season** run series win Astros player hit league start lead goal think Tortorella tampa score world
5	stone Ruiz Toney **fight show** jagger world rolling **play win** song know want title **champion** mick tour game band heavyweight **boxing**

Table 5.13: Characteristic words of context clusters inside the thematic-specific reference for the relation *lead_by*

concentrated mainly on external measures by comparing their results with our reference. Basic clustering with POS weighting was proved to be more effective than binary weighting or tf-idf weighing, providing very precise basic clusters. Experiments also showed that the clustering refinement procedure for basic clustering can improve *recall* and keep the same level of *precision*. In semantic clustering experiments, we found that corpus-based similarities achieved better results than WordNet-based similarities. Moreover, the comparison of results for different categories of words illustrates that the verbs are the most important words to group relation instances semantically but the nouns provide complementary information to refine the clustering. Additionally, the analysis of similarity distribution against the reference confirms the advantage of the proposed multi-level relation clustering method.

At last, our experiments showed that context clustering succeeded at grouping thematic segments into different themes. The topic-based relation clustering was demonstrated to be useful to form highly precise relation clusters with a reasonable recall by focusing on the intersection between relation clusters and context clusters. These results were particularly evaluated by a specific reference of relation clusters involving different themes. A significant improvement of *purity* and *NMI* were observed. This improvement comes with a drop of recall, which could be solved with a better context clustering.

Some perspectives can be considered. First, in our current work, semantic clustering is mainly based on the $Cmid$ part of the relation mention. Indeed, most of the important information for characterizing a relation is present in the $Cmid$ part of the mention according to investigations of other work (Ebadat, 2012). Nevertheless, complementary information from the $Cpost$ part can sometimes be interesting as well, such as in the following extracted instances for the relation *accuse* (category PER−PER):

- e.g. "Kerry accused President Bush of shattering alliances, misleading Americans, ignoring his own advisers, failing to plan for the peace, creating a haven for terrorists in Iraq and ignoring the tragic consequences of his 'colossal misjudgments'."

- e.g. "Thomas Boswell publicly accused Jose Canseco of steroid use, but he had to be aware of it, to say nothing of the rumors flying around the game."

- e.g. "Brett Favre, who criticized receiver Javon Walker for his attempts to renegotiate his contract this off-season, McNabb has stayed out of Owens' contract squabble."

All these three examples are about the *accusation* of one person by another person. However, rather than considering only the act of *accusing* itself, relation clustering should

also consider the content of each accusation and group similar contents together in a specific context. In this case, $Cpost$ bears a complementary but important information.

The topic-based relation clustering is still a preliminary research in our current work. More researches can be performed first for the context clustering itself. Based on the results of our relation clustering experiments, a second level of context clustering could be applied to form larger context clusters. Moreover, efficient clustering algorithms such as K-means clustering are potentially suitable for this second level of context clustering since a predefined number of themes is not an obstacle for our topic-based relation clustering and can avoid the production of too fine-grained context clusters. Finally, additional methods for integrating thematic information into relation clusters could be explored.

Chapter 6

Conclusions and Perspectives

6.1 A Brief Conclusion on Contributions

In this thesis, we were interested in the issue of relation extraction and relation clustering in the context of unsupervised Information Extraction, with more particularly a focus on binary relations between named entities in large open domain corpora. Our objective was to find automatically instances of relations between named entity pairs from raw text input such as newspapers, and then to group certain relation instances together according to their similarities so that each relation instance can be characterized by its corresponding cluster. This issue raises many difficulties:

- How to find these relation instances from documents in open domain?
- How to guarantee the existence of a valid relation between a pair of named entities?
- How to decide the similarity between relation instances?
- How to cluster relation instances when the considered corpus is massive?
- How to evaluate the resulting clusters when even no direct relevant reference is available?

This thesis has designed a whole pipeline to tackle these problems. Our experiments started from a 18-month newspaper articles from the *New York Times* newspaper and led to extract 165,708 reliable relation instances between named entity types including *person*, *organization* and *location* (some samples are shown in Figure 3.7, Page 65). These relation instances were then clustered in different ways, either by their semantic similarity or by their topical similarity. For the former, one of our stable and most performing results

generated 10,116 different semantic clusters (statistics shown In Table 5.9, Page 134), each of which contains different ways of expressing a type of relations (samples shown in Table 5.10, Page 135). Different aspects of the contributions in this thesis is outlined in the following paragraphs.

Relation prototype for candidate extraction

In the beginning of this thesis, we proposed a relation prototype which is consisted of three main elements: arguments, mention and context. The arguments are the named entities between that are linked by the relation. The mention is how this relation is expressed. It is more precisely divided into three parts – $Cpre$, $Cmid$, $Cpost$ – according to their position around the arguments. The context is the thematic segment in which a relation instance occurs. It provides a more global information than the first two elements. Starting from this relation prototype, we extract relation instances by searching in the scope of a sentence pairs of named entities separated by at least one verb so that candidates for relation instances can be obtained efficiently, involving an open variety of topics and semantic meanings.

Filtering to get rid of false relation instances

Once a large quantity of relation instances candidates are available, we specified a two-step filtering procedure to discard false relations in order to be more confirmed for the existence of valid relations in extracted candidates. Based on our analysis of extracted relation instances candidates, we first defined filtering heuristics and applied them to remove a large amount of false relations efficiently. Then, a second step of filtering based on machine learning models was implemented to refine the selection of reliable relation instances. We demonstrated that our statistical models, even trained without deep linguistic features such as syntactic relation, can help detecting valid relations, achieving both satisfying *recall* and *precision*. Our best trained model achieves the performance that 76.2% of extracted relation instances are valid ones.

Clustering to organize instances in different ways

Compared to the extraction of instances with predefined relation types in traditional Information Extraction systems, the proposed approach is in the context of Unsupervised Information Extraction so that it puts no constraints on the relation types and therefore can

discover unknown relations from unstructured text. Moreover, our approach goes further than most of the existing UIE systems by investigating different methods for organizing the extracted relation instances rather than merely extracting them.

We explored different ways of organizing the extracted relation instances according to their semantic meanings. In order to face the computation difficulty using big data, we proposed a multi-level clustering procedure to group relation instances in two steps. In the first step, named basic clustering, we focused on grouping relation instances that are characterized by the same words, with only some variations about the way they are expressed. We proved that the association of a weighting strategy based on part-of-speech categories and the *Cosine* similarity measure between relation instances led to form very precise basic clusters, achieving 0.81 as *precision* in our experiments. In the second step, called semantic clustering, we are interested at investigating several semantic similarity measures, either WordNet-based measures or corpus-based measures, at three levels of granularity – word, relation instance and basic cluster – in order to group similar basic clusters whose relation instances are expressed by synonyms or more complex paraphrases. Our experiments have showed that this second step of clustering is able to handle complex linguistic phenomena so that the *recall* of result clusters can be improved. The best results of semantic clustering was obtained by a corpus-based similarity, with the *recall* measure arising from 0.40 in basic clusters to 0.54 in semantic clusters.

In addition to using only local information coming from sentences, we also included thematic information to perform a topic-based relation clustering. A context clustering was first applied to group similar thematic segments so that the contexts of relation instances are organized into different clusters, each of which represents a specific theme. Our experiments regarding the integration of such context clustering and relation clustering have demonstrated that thematic information can not only provide a thematic view of relation instances but can also be used to improve the quality of semantic clusters.

Clustering evaluation and the reference

For the clustering evaluation issues, we first investigated how internal measures can be used to evaluate the quality of clustering when no reference is available for specific tasks carries out. Furthermore, we designed a method for building references semi-automatically, supported by a tool combining a browser based interface and a search engine based query procedure. We have completed an annotation of 80 semantic relation clusters containing 4,420 relation instances, which is one of the largest references in current unsupervised IE community for relation clustering tasks. With the help of this reference, external evaluation

measures at different levels (*i.e.* relation instance level and cluster level) were implemented and used for evaluating the results of our different clustering methods.

6.2 What Can Be Done Next?

The system presented in this thesis succeeds at extracting and clustering unknown relation instances at a large scale. Nevertheless, many aspects can be improved or extended and some issues are still open both for relation extraction and relation clustering. We give several possible perspectives below.

Perspectives for relation extraction

A relation instance in our prototype contains three elements: arguments, mention, context. The first two elements are essential to determine the validity of the relation. During our initial extraction of relation candidates, we require the presence of at least one verb in the $Cmid$ part of the mention. However, this constraint could be fully removed to include more candidates, considering that we are working in a context of open domain where expressions are much more diverse than in specific domains and there also exists many relations expressed using only nouns or adjectives. The heuristics that we defined for a preliminary and efficient removal of a large amount of false relations as the first step of filtering, could also be largely extended. Existing work about rule-based systems for relation extraction is a sources of ideas for enriching these heuristics. The machine learning models for the second step of filtering are even more flexible for improvements, by testing different ways of representing relation instances and different features for training.

Perspectives for relation clustering

In our work about relation clustering, our observations tell us that the $Cmid$ part of the mention of relation instances often refers to the most important information of the corresponding semantic relation. Therefore, only the $Cmid$ part is used to calculate semantic similarities between relation instances. However, the $Cpost$ part of the mention may sometimes contain important information as we have already discussed in the previous chapter. Moreover, each part of this representation is currently a unigram model and a richer representation using n-grams could be considered for experiments. Finally, the arguments themselves could be considered as a criterion for clustering as well to verify all possible relations between similar pairs of arguments.

Topic-based relation clustering, which considers the integration of thematic information into the organization of semantic relations is rather new in the field of unsupervised IE and is worthy to be continued for investigating different methods to extract this thematic information and different ways to use it.

The current results of semantic clustering can be viewed as a three-level hierarchy: relation category, semantic cluster and basic cluster following a top-down order. For the results of topic-based relation clustering, we have also a three-level hierarchy: relation category, semantic cluster and theme-specific relation cluster. Taking into account more elements for clustering, such as relation arguments or the $Cpost$ part of the mentions, can lead to define a more complex hierarchical organization. However, according to our experiments about the sequential application of relation clustering and context clustering, the result clusters might be too small if the clustering processes based on these different elements (i.e. $Cmid$, $Cpost$, arguments and context) are applied one after another. Therefore, a soft clustering method may be more suitable to include a large set of dimensions for relation clustering. Moreover, the evaluation of such multi-dimensional clustering is also a difficult issue. For example, one limit of the evaluation of our results is that semantic clusters and basic clusters are only evaluated at the semantic cluster level. Consequently, it might be necessary to update our reference to make it adapted to new evaluation tasks.

Separation or unification: relation extraction and relation clustering

As stated previously, we treated relation extraction and relation clustering as two separated tasks. It makes the procedure of relation extraction more efficient and less dependent on the corpus size compared to systems using clustering algorithms for both tasks at the same time. Moreover, this separation makes the relation clustering procedure more reliable since relation instances are preselected by the relation extraction procedure. However, this separation does not prevent from joining the two tasks in another way. The statistical models used for determining the validity of a relation are often trained by relying on features around its arguments such as the sequences of part-of-speech. However, the validity of a relation based on such linguistic criteria does not guarantee that it is actually a concrete relation in the real world. Hence, it could be interesting to exploit information from the results of relation clustering at the relation extraction stage by integrating it in the machine learning models defined for such extraction.

Appendix A

Overall Performance Estimation of Two-step Filtering

A.1 Contingency Table

A contingency table is a type of table in matrix format which illustrates the frequency distribution of different variables. In the case of the evaluation of classifier, this frequency distribution is based on the number of different decisions of the classifier.

When a classifier is applied to a set of test data, there exists generally four kind of decisions. Two kinds of them are correct ones: a *True Positive* (**TP**) decision gives a positive answer to a true test example while a *True Negative* (**TN**) decision gives a negative answer to a false test example. On the other side, there are two kinds of incorrect decisions: a *False Negative* (**FN**) decision annotate a true test example as negative, and a *False Positive* (**FP**) decision considers a false example as positive. Consequently, the contingency table for classifier evaluation based on the distribution of these four types of decisions can be illustrated as in Table A.1.

	$Positive$	$Negative$
True	TP	FN
False	FP	TN

Table A.1: Contingency table for classifier evaluation

With the definition of four types of decisions by the classifier, F-measures can be calculated from the contingency table:

$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN}, \quad F_\beta = \frac{(\beta^2 + 1) \cdot P \cdot R}{\beta^2 \cdot P + R} \tag{A.1}$$

Two-level Contingency Table In certain experiments, different classifiers could be applied sequentially to test data, for example the second classifier examines the decisions for the positive decisions given by the first classifier. In this case, a new contingency table can be built for the second classifier. However, if one is interested by the overall performance of these two classifiers, a two-level contingency table could be given such as illustrated in Table A.2:

	$Positive$		$Negative$
	$Positive'$	$Negative'$	
True	TP'	FN'	FN
False	FP'	TN'	TN

Table A.2: Two-level contingency table

The F-measure evaluation for the second classifier can be calculated by:

$$P' = \frac{TP'}{TP' + FP'}, \quad R' = \frac{TP'}{TP' + FN'}, \quad F'_\beta = \frac{(\beta^2 + 1) \cdot P' \cdot R'}{\beta^2 \cdot P' + R'} \tag{A.2}$$

Meanwhile, the overall performance of two classifiers should be:

$$P^* = \frac{TP'}{TP' + FP'}, \quad R^* = \frac{TP'}{TP' + FN' + FN}, \quad F^*_\beta = \frac{(\beta^2 + 1) \cdot P^* \cdot R^*}{\beta^2 \cdot P^* + R^*} \tag{A.3}$$

It is clear to observe that the value of precision measure stays the same with the performance of the second classifier, whereas, the recall measure should be recalculated. In the circumstance of relation filtering in this thesis, the filtering heuristics can be considered as the first classifier, while the filtering with statistical model is the second step of classifier. The overall F-measures performance can be approximated using the performance and the statistics of each filtering step.

A.2 Approximation of Overall F-measures

Before estimating overall F-measures of two-step filtering on relation candidates, two definitions of measures are proposed as the following:

- Conservation ratio ($\mathbf{C}$) : the proportion of relation candidates kept by filtering procedures, which could be the filtering heuristics, statistical filtering or both

- True relation ratio ($\mathbf{T}$) : the proportion of true relation candidates among all relation candidates tested, either by filtering heuristics or by statistical models

According to the definitions, we can have the measure $\mathbf{C}$ for the filtering heuristics and $\mathbf{C}^*$ for two-steps filtering, and also the measures $\mathbf{T}$ and $\mathbf{T}'$ for respectively filtering heuristics and statistical filtering by calculating:

$$C = \frac{TP+FP}{TP+FN+FP+TN} = \frac{TP'+FN'+FP'+TN'}{TP+FN+FP+TN} \tag{A.4}$$

$$C^* = \frac{TP'+FP'}{TP+FN+FP+TN} = \frac{TP'+FP'}{TP'+FP'+FN'+TN'+FN+TN} \tag{A.5}$$

$$T = \frac{TP+FN}{TP+FN+FP+TN} \tag{A.6}$$

$$T' = \frac{TP'+FN'}{TP'+FN'+FP'+TN'} \tag{A.7}$$

With all definitions from A.1 to A.7, and also the fact of:

$$TP = TP'+FN' \tag{A.8}$$

$$FP = FP'+TN' \tag{A.9}$$

the overall recall can be calculated as:

$$
\begin{aligned}
R^* &= \frac{TP'}{TP' + FN' + FN} && \text{(A.10)}\\
&= \frac{R'(TP' + FN')}{TP + FN} && \text{(A.11)}\\
&= \frac{R'(TP' + FN')}{T(TP + FN + FP + TN)} && \text{(A.12)}\\
&= \frac{R'T'(TP' + FN' + FP' + TN')}{T(TP + FN + FP + TN)} && \text{(A.13)}\\
&= \frac{R'T'C}{T} && \text{(A.14)}
\end{aligned}
$$

Calculation of C Conservation ratio C is calculated by counting numbers of all four kinds of decisions by heuristics, which is 47.2%, according to Table 3.12. In the same way, the conservation ratio of two-step filtering C^* is given as 28.6%.

Estimation of T and T' Table 3.3 show the results about the manual annotation of the true relation ratio for both the kept and filtered candidates by the filtering heuristics. For each of the six relation types considered, 50 random relation candidates are chosen and then numbers of true relation are counted for both kept part and filtered part, respectively TP_i^{kept} and $FN_i^{filtered}$.

Therefore, for each relation type, we can have the true relation ratio of initially extracted candidates T_i and that of kept candidates by filtering heuristics T_i'

$$
\begin{aligned}
T_i &= \frac{TP + FN}{TP + FN + FP + TN} \simeq \frac{TP_i^{kept} + FN_i^{filtered}}{100} && \text{(A.15)}\\
T_i' &= \frac{TP' + FN'}{TP' + FN' + FP' + TN'} \simeq \frac{TP_i^{kept}}{50} && \text{(A.16)}
\end{aligned}
$$

Since the volume for each type of relation are not the same, the total measure T or T' for six types of relations considered are given by the sum of the measure for each relation type with a weight. This weight is calculated directly by the proportion of the volume of one relation type on that of all relation types, given by Table 3.1. True relation ratio for each relation type is calculated and then illustrated in Table A.3.

Relation type	$\mathbf{TP^{kept}}$	$\mathbf{FN^{filtered}}$	$\mathbf{T_i}$ (%)	$\mathbf{T'_i}$ (%)	**Weight**(%)
ORG – LOC	14	7	2.23	2.97	10.61
ORG – ORG	20	6	2.96	4.55	11.37
ORG – PER	20	4	2.62	4.36	10.91
PER – LOC	40	13	11.93	18.01	22.52
PER – ORG	40	12	9.69	14.91	18.64
PER – PER	14	5	4.93	7.27	25.95
ALL	148	47	34.36	52.08	100

Table A.3: Estimation of T and T'

The total ratio for all relation types can be calculated as:

$$T = \sum_i T_i \, Weight_i = 34.36\% \tag{A.17}$$

$$T' = \sum_i T'_i \, Weight_i = 52.08\% \tag{A.18}$$

Estimation of overall recall Since sequential model with CRF is finally chosen for the statistical model filtering, the overall recall can be calculated with Equation A.14, using the results given in Table 3.7:

$$R^* = \frac{R'T'C}{T} \tag{A.19}$$

$$= R' * \frac{0.5208 * 0.472}{0.3436} \tag{A.20}$$

$$= R' * 0.715 \tag{A.21}$$

$$= 0.559 \tag{A.22}$$

The overall F-measure for two-step filtering can be obtained as well. The whole results are illustrated in Table A.4

Model	**Accuracy**	**Precision**	**Recall**	**F1-measure**
CRF	0.745	0.762	0.782	0.771
TOTAL	/	0.762	0.559	0.645
Banko and Etzioni (2008)	/	0.883	0.452	0.598

Table A.4: Overall F-measures estimation for two-step filtering

A.3 Estimation of The Ratio of False Negative Decisions

On the other side of the performance evaluation of two-step filtering, it is also important to know how many true relations are rejected by filtering procedure. The ratio (T_{FN}) can be defined as:

$$T_{FN} = \frac{FN + FN'}{FN + FN' + TN + TN'} \tag{A.23}$$

The value of $FN + FN'$ can be calculated by definition of recall in Equation A.3:

$$FN + FN' = (\frac{1}{R} - 1) * TP' \tag{A.24}$$

While the value of $FN + FN' + TN + TN'$ can be calculated by the definition of conservation ratio $\mathbf{C}^*$ in Equation A.5:

$$FN + FN' + TN + TN' = (\frac{1}{C^*} - 1) * (TP + TP') \tag{A.25}$$

The value of T_{FN} can be easily calculated as:

$$\begin{aligned} T_{FN} &= \frac{FN + FN'}{FN + FN' + TN + TN'} && \text{(A.26)} \\ &= \frac{(\frac{1}{R} - 1) * TP'}{(\frac{1}{C^*} - 1) * (TP + TP')} && \text{(A.27)} \\ &= \frac{\frac{1}{R} - 1}{\frac{1}{C^*} - 1} * P && \text{(A.28)} \\ &= \frac{0.789}{0.250} * 0.762 && \text{(A.29)} \\ &= 0.240 && \text{(A.30)} \end{aligned}$$

Therefore, among all filtered relation instances, the true relation ratio is estimated to be 24.0%, which is acceptable for an unsupervised relation extraction work in open domain.

Publications

Jean-Louis, L., Besançon, R., Ferret, O., and Wang, W. (2011). Using a weakly supervised approach and lexical patterns for the kbp slot filling task. In *4th Text Analysis Conference (TAC2011)*, Gaithersburg, Maryland, USA.

Wang, W., Besançon, R., Ferret, O., and Grau, B. (2011a). Filtering and clustering relations for unsupervised information extraction in open domain. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, pages 1405–1414, New York, NY, USA. ACM.

Wang, W., Besançon, R., Ferret, O., and Grau, B. (2011b). Filtrage de relations pour l'extraction d'information non supervisée. In *18ème Conférence sur le Traitement Automatique des Langues Naturelles, session articles courts*, TALN 2011, Montpellier, France.

Wang, W., Besançon, R., Ferret, O., and Grau, B. (2012a). Evaluation of unsupervised information extraction. In *Proceedings of the Eight International Conference on Language Resources and Evaluation*, LREC'12, Istanbul, Turkey. European Language Resources Association (ELRA).

Wang, W., Besançon, R., Ferret, O., and Grau, B. (2012b). Regroupement de relations pour l'extraction d'information non supervisée. In *Proceedings of the 9th French Information Retrieval Conference (Conférence en Recherche d'Infomations et Applications)*, CORIA 2012, Bordeaux, France.

Wang, W., Besançon, R., Ferret, O., and Grau, B. (2013). Regroupement sémantique de relations pour l'extraction d'information non supervisée. In *20ème Conférence sur le Traitement Automatique des Langues Naturelles*, TALN 2013, Les Sables-d'Olonne, France.

Bibliography

Agichtein, E. and Gravano, L. (2000). Snowball: extracting relations from large plain-text collections. In *Proceedings of the fifth ACM conference on Digital libraries*, DL '00, pages 85–94, New York, NY, USA. ACM.

Akbik, A. and Broß, J. (2009). Extracting semantic relations from natural language text using dependency grammar patterns. In *Proceedings of the Workshop on Semantic Search (SemSearch 2009) at the 18th International World Wide Web Conference*, WWW 2009, Madrid, Spain.

Akbik, A. and Löser, A. (2012). Kraken: N-ary facts in open information extraction. In *The Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, AKBC-WEKEX 2012, Montréal, Canada.

Amigó, E., Gonzalo, J., Artiles, J., and Verdejo, F. (2009). A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*, 12(4):461–486.

Anderberg, M. (1973). *Cluster analysis for applications*. Probability and mathematical statistics. Academic Press.

Aone, C., Halverson, L., Hampton, T., and Ramos-Santacruz, M. (1998). Sra: Description of the ie2 system used for muc-7. In *In Proceedings of Seventh Message Understanding Conference*, MUC7, Virginia, USA.

Balog, K., de Vries, A. P., Serdyukov, P., Thomas, P., and Westerveld, T. (2010). Overview of the trec 2009 entity track. In *Proceedings of the Eighteenth Text REtrieval Conference*, TREC 2009. NIST.

Balog, K., Serdyukov, P., and de Vries, A. P. (2011). Overview of the trec 2010 entity track. In *Proceedings of the Nineteenth Text REtrieval Conference*, TREC 2010. NIST.

Balog, K., Serdyukov, P., and de Vries, A. P. (2012). Overview of the trec 2011 entity track. In *Proceedings of the Twentieth Text REtrieval Conference*, TREC 2011. NIST.

Banko, M. (2009). *Open information extraction from the web*. PhD thesis, University of Washington.

Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., and Etzioni, O. (2007). Open information extraction from the web. In *Proceedings of the 20th international joint conference on Artifical intelligence*, IJCAI'07, pages 2670–2676, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Banko, M. and Etzioni, O. (2007). Strategies for lifelong knowledge extraction from the web. In *Proceedings of the 4th international conference on Knowledge capture*, K-CAP '07, pages 95–102, New York, NY, USA. ACM.

Banko, M. and Etzioni, O. (2008). The tradeoffs between open and traditional relation extraction. In *Proceedings of ACL-08: HLT*, pages 28–36, Columbus, Ohio. Association for Computational Linguistics.

Bayardo, R. J., Ma, Y., and Srikant, R. (2007). Scaling up all pairs similarity search. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 131–140, New York, NY, USA. ACM.

Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., and Hellmann, S. (2009). Dbpedia - a crystallization point for the web of data. *Web Semant.*, 7(3):154–165.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, SIGMOD '08, pages 1247–1250, New York, NY, USA. ACM.

Bollegala, D. T., Matsuo, Y., and Ishizuka, M. (2010). Relational duality: unsupervised extraction of semantic relations between entities on the web. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 151–160, New York, NY, USA. ACM.

Brady, D., Childs, L., Cassel, D., Magee, B., Heintzelman, N., and Weir, C. (1998). Description of lockheed martin's nltoolset as applied to muc-7. In *Proceedings of the Seventh Message Understanding Conference*, MUC-7, Fairfax, Virginia.

Brin, S. (1998). Extracting patterns and relations from the world wide web. In *Selected papers from the International Workshop on The World Wide Web and Databases*, WebDB '98, pages 172–183, London, UK, UK. Springer-Verlag.

Broder, A. Z., Glassman, S. C., Manasse, M. S., and Zweig, G. (1997). Syntactic clustering of the web. In *Selected papers from the sixth international conference on World Wide Web*, pages 1157–1166, Essex, UK. Elsevier Science Publishers Ltd.

Bunescu, R. C. and Mooney, R. J. (2005). A shortest path dependency kernel for relation extraction. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 724–731, Stroudsburg, PA, USA. Association for Computational Linguistics.

Cafarella, M. J., Downey, D., Soderland, S., and Etzioni, O. (2005). Knowitnow: fast, scalable information extraction from the web. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 563–570, Stroudsburg, PA, USA. Association for Computational Linguistics.

Campion, N., Closson, J., O., F., Shin, J., Grau, B., Lazard, J. M., LAHBIB, D., Besançon, R., FLORET, J.-M., and MEZAOUR, A.-D.and TANNIER, X. (2010). Filtrar-s : Un outil de filtrage sémantique et de fouille de textes pour la veille. In *Actes du colloque : Veille stratégique scientifique et technique*, VSST' 2010, Toulouse. Université Paul Sabatier.

Carlson, A., Betteridge, J., Wang, R. C., Hruschka, Jr., E. R., and Mitchell, T. M. (2010). Coupled semi-supervised learning for information extraction. In *Proceedings of the third ACM international conference on Web search and data mining*, WSDM '10, pages 101–110, New York, NY, USA. ACM.

Chambers, N. and Jurafsky, D. (2011). Template-based information extraction without the templates. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, HLT '11, pages 976–986, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chen, J., Ji, D., Tan, C., and Niu, Z. (2005). Unsupervised feature selection for relation extraction. In *In Proceedings of the Second International Joint Conference on Natural Language Processing*, pages 262–267. Proceedings of IJCNLP-2005.

Cheu, E. Y., Kwoh, C. K., and Zhou, Z. (2004). On the two-level hybrid clustering algorithm. In *The International Conference on Artificial Intelligence in Science and Technology*, AISAT'04, pages 138–142.

Ciaramita, M., Gangemi, A., Ratsch, E., Šaric, J., and Rojas, I. (2005). Unsupervised learning of semantic relations between concepts of a molecular biology ontology. In *Proceedings of the 19th international joint conference on Artificial intelligence*, IJCAI'05, pages 659–664, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Culotta, A., McCallum, A., and Betz, J. (2006). Integrating probabilistic extraction models and data mining to discover relations and patterns in text. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL '06, pages 296–303, Stroudsburg, PA, USA. Association for Computational Linguistics.

Culotta, A. and Sorensen, J. (2004). Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Barcelona, Spain. Association for Computational Linguistics.

De Marneffe, M.-C., MacCartney, B., and Manning, C. D. (2006). Generating typed dependency parses from phrase structure trees. In *The 5th international conference on Language Resources and Evaluation*, LREC 2006, pages 449–454, Genoa, Italy. European Language Resources Association (ELRA).

Dias, G., Alves, E., and Lopes, J. G. P. (2007). Topic segmentation algorithms for text summarization and passage retrieval: An exhaustive evaluation. In *Proceedings of the 22Nd National Conference on Artificial Intelligence - Volume 2*, AAAI'07, pages 1334–1339. AAAI Press.

Doddington, G., Mitchell, A., Przybocki, M., Ramshaw, L., Strassel, S., and Weischedel, R. (2004). The automatic content extraction (ace) program–tasks, data, and evaluation. In *The 4th Proceedings of Conference on Language Resources and Evaluation*, LREC 2004, pages 837–840, Lisbon, Portugal. European Language Resources Association (ELRA).

Dolan, B., Quirk, C., and Brockett, C. (2004). Unsupervised construction of large paraphrase corpora: exploiting massively parallel news sources. In *Proceedings of the 20th international conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Downey, D., Etzioni, O., and Soderland, S. (2005). A probabilistic model of redundancy in information extraction. In *Proceedings of the 19th international joint conference on Artificial intelligence*, IJCAI'05, pages 1034–1041, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Downey, D., Etzioni, O., and Soderland, S. (2010). Analysis of a probabilistic model of redundancy in unsupervised information extraction. *Artificial Intelligence*, 174(11):726–748.

Ebadat, A. R. (2012). *Toward Robust Information Extraction Models for Multimedia Documents*. PhD thesis, Université Européenne de Bretagne.

Eichler, K., Hemsen, H., and Neumann, G. (2008). Unsupervised relation extraction from web documents. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Embarek, M. and Ferret, O. (2008). Learning patterns for building resources about semantic relations in the medical domain. In *6^{th} Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco.

Ertöz, L., Steinbach, M., and Kumar, V. (2001). Finding topics in collections of documents: A shared nearest neighbor approach. In *In Proceedings of Text Mine'01, First SIAM International Conference on Data Mining*.

Ertöz, L., Steinbach, M., and Kumar, V. (2002). A new shared nearest neighbor clustering algorithm and its applications. In *Workshop on Clustering High Dimensional Data and its Applications at 2nd SIAM International Conference on Data Mining*.

Ertöz, L., Steinbach, M., and Kumar, V. (2003). Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. In *Proceedings of the Third SIAM International Conference on Data Mining (SDM 2003)*, Proceedings in Applied Mathematics. Society for Industrial and Applied Mathematics.

Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D. S., and Yates, A. (2004a). Web-scale information extraction in knowitall: (preliminary results). In *Proceedings of the 13th international conference on World Wide Web*, WWW '04, pages 100–110, New York, NY, USA. ACM.

Etzioni, O., Cafarella, M., Downey, D., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D. S., and Yates, A. (2004b). Methods for domain-independent information extraction from the web: an experimental comparison. In *Proceedings of the 19th national conference on Artifical intelligence*, AAAI'04, pages 391–398. AAAI Press.

Etzioni, O., Cafarella, M., Downey, D., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D. S., and Yates, A. (2005). Unsupervised named-entity extraction from the web: an experimental study. *Artifical intelligence*, 165:91–134.

Etzioni, O., Fader, A., Christensen, J., Soderland, S., and Mausam, M. (2011). Open information extraction: the second generation. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence*, IJCAI'11, pages 3–10. AAAI Press.

Fader, A., Soderland, S., and Etzioni, O. (2011). Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1535–1545, Stroudsburg, PA, USA. Association for Computational Linguistics.

Feldman, R. and Rozenfeld, B. (2006). Boosting unsupervised relation extraction by using ner. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, page 473, Morristown, NJ, USA. Association for Computational Linguistics.

Ferret, O. (2004). Discovering word senses from a network of lexical cooccurrences. In *Proceedings of the 20th international conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ferret, O. (2007). Finding document topics for improving topic segmentation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, ACL '07, Prague, Czech Republic. The Association for Computer Linguistics.

Ferret, O. (2010). Testing semantic similarity measures for extracting synonyms from a corpus. In *Proceedings of the International Conference on Language Resources and Evaluation*, LREC'10, Valletta, Malta. European Language Resources Association.

Firth, J. R. (1957). *A synopsis of linguistic theory 1930-1955*, volume 1952-59. The Philological Society, Oxford.

Fung, G. P. C., Yu, J. X., and Lu, H. (2002). Discriminative category matching: Efficient text classification for huge discriminative category matching. In *Proceedings of the 2002 IEEE International Conference on Data Mining*, ICDM '02, pages 187–194, Washington, DC, USA. IEEE Computer Society.

Galley, M., McKeown, K., Fosler-Lussier, E., and Jing, H. (2003). Discourse segmentation of multi-party conversation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, ACL '03, pages 562–569, Sapporo, Japan. Association for Computational Linguistics.

Gamallo, P., Garcia, M., and Fernández-Lanza, S. (2012). Dependency-based open information extraction. In *Proceedings of the Joint Workshop on Unsupervised and Semi-Supervised Learning in NLP*, pages 10–18, Avignon, France. Association for Computational Linguistics.

Gonzàlez, E. and Turmo, J. (2009). Unsupervised relation extraction by massive clustering. In *Proceedings of the 2009 Ninth IEEE International Conference on Data Mining*, ICDM '09, pages 782–787, Washington, DC, USA. IEEE Computer Society.

Grishman, R. (2012). Structural linguistics and unsupervised information extraction. In *the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, AKBC-WEKEX 2012, Montréal, Canada.

Grishman, R. and Min, B. (2010). New york university kbp 2010 slot-filling system. In *Text Analysis Conference (TAC)*. NIST.

Grishman, R. and Sundheim, B. (1996). Message understanding conference-6: a brief history. In *Proceedings of the 16th conference on Computational linguistics*, COLING '96, pages 466–471, Stroudsburg, PA, USA. Association for Computational Linguistics.

Guinaudeau, C., Gravier, G., and Sébillot, P. (2012). Enhancing lexical cohesion measure with confidence measures, semantic relations and language model interpolation for multimedia spoken content topic segmentation. *Computer Speech and Language*, 26(2):90–104.

Halkidi, M., Batistakis, Y., and Vazirgiannis, M. (2002). Cluster validity methods: part i. *SIGMOD Record*, 31(2):40–45.

Handl, J., Knowles, J., and Kell, D. B. (2005). Computational cluster validation in post-genomic data analysis. *Bioinformatics*, 21:3201–3212.

Hasegawa, T., Sekine, S., and Grishman, R. (2004). Discovering relations among named entities from large corpora. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Barcelona, Spain. Association for Computational Linguistics.

Hearst, M. A. (1997). Texttiling: segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64.

Huyck, C. R. (1998). American university in cairo: Description of the american university in cairo's system used for muc-7. In *Seventh Message Understanding Conference*, MUC-7, Fairfax, Virginia.

Jain, A. K. and Dubes, R. C. (1988). *Algorithms for clustering data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323.

Jarvis, R. A. and Patrick, E. A. (1973). Clustering using a similarity measure based on shared near neighbors. *IEEE Transactions on Computers*, 22(11):1025–1034.

Jean-Louis, L., Besançon, R., Ferret, O., and Wang, W. (2011). Using a weakly supervised approach and lexical patterns for the kbp slot filling task. In *4th Text Analysis Conference (TAC2011)*, Gaithersburg, Maryland, USA.

Ji, H. and Grishman, R. (2011). Knowledge base population: successful approaches and challenges. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, HLT '11, pages 1148–1158, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jiang, J. and Conrath, D. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the International Conference on Research in Computational Linguistics*, pages 19–33.

Joachims, T. (1999). Making large-scale SVM learning practical. In *Advances in Kernel Methods - Support Vector Learning*, pages 169–184. MIT Press, Cambridge, MA.

Kambhatla, N. (2004). Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, ACLdemo '04, Barcelona, Spain. Association for Computational Linguistics.

Kok, S. and Domingos, P. (2008). Extracting semantic networks from text via relational clustering. In *Proceedings of the 2008 European Conference on Machine Learning and Knowledge Discovery in Databases - Part I*, ECML PKDD '08, pages 624–639, Berlin, Heidelberg. Springer-Verlag.

Kurita, T. (1991). An efficient agglomerative clustering algorithm using a heap. *Pattern Recognition*, 24(3):205–209.

Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Lavergne, T., Cappé, O., and Yvon, F. (2010). Practical very large scale crfs. In *48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 504–513, Uppsala, Sweden.

Leacock, C. and Chodorow, M. (1998). *Combining local context and WordNet similarity for word sense identification*, pages 305–332. In C. Fellbaum (Ed.), MIT Press.

Lenat, D. B. (1995). Cyc: a large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38.

Levenshtein, V. (1966). Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10:707.

Lin, D. (1998a). Automatic retrieval and clustering of similar words. In *Proceedings of the 17th international conference on Computational linguistics*, COLING '98, pages 768–774, Stroudsburg, PA, USA. Association for Computational Linguistics.

Lin, D. (1998b). An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 296–304, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297. University of California Press.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B. (1993). Building a large annotated corpus of english: the penn treebank. *Computational Linguistics*, 19(2):313–330.

McCallum, A. K. (2002). Mallet: A machine learning for language toolkit. `http://mallet.cs.umass.edu`.

McDonald, R., Pereira, F., Kulick, S., Winters, S., Jin, Y., and White, P. (2005). Simple algorithms for complex relation extraction with applications to biomedical ie. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 491–498, Ann Arbor, Michigan. Association for Computational Linguistics.

Meyers, A., Grishman, R., Kosaka, M., and Zhao, S. (2001). Covering treebanks with glarf. In *Proceedings of the ACL 2001 Workshop on Sharing Tools and Resources - Volume 15*, STAR '01, pages 51–58, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mihalcea, R., Corley, C., and Strapparava, C. (2006). Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the 21st national conference on Artificial intelligence - Volume 1*, AAAI'06, pages 775–780. AAAI Press.

Miller, G. A. (1995). Wordnet: a lexical database for english. *Commun. ACM*, 38(11):39–41.

Miller, S., Crystal, M., Fox, H., Ramshaw, L., Schwartz, R., Stone, R., Weischedel, R., and Group, T. A. (1998). Algorithms that learn to extract information bbn: Description of the sift system as used for muc-7. In *In Proceedings of Seventh Message Understanding Conference*, MUC7, Virginia, USA.

Min, B. and Grishman, R. (2012). Challenges in the knowledge base population slot filling task. In *Proceedings of the Eight International Conference on Language Resources and Evaluation*, LREC 2012, Istanbul, Turkey. European Language Resources Association (ELRA).

Min, B., Shi, S., Grishman, R., and Lin, C.-Y. (2012). Ensemble semantics for large-scale unsupervised relation extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '12, Stroudsburg, PA, USA. Association for Computational Linguistics.

Minard, A.-L. (2012). *Extraction de relations en domaine de spécialité*. PhD thesis, Université de Paris-Sud.

Mintz, M., Bills, S., Snow, R., and Jurafsky, D. (2009). Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, ACL '09, pages 1003–1011, Suntec, Singapore. Association for Computational Linguistics.

MUC (1992). *MUC4 '92: Proceedings of the 4th conference on Message understanding*, McLean, Virginia, USA. Association for Computational Linguistics.

MUC (1995). *MUC6 '95: Proceedings of the 6th conference on Message understanding*, Stroudsburg, PA, USA. Association for Computational Linguistics.

MUC (1998). *MUC7 '98: Proceedings of the Seventh Message Understanding Conference*, Fairfax, Virginia, USA. Association for Computational Linguistics.

Pareti, S. (2012). A database of attribution relations. In *Proceedings of the Eight International Conference on Language Resources and Evaluation*, LREC 2012, Istanbul, Turkey. European Language Resources Association (ELRA).

Patten, T., Hoffman, B., and Thurn, M. (1998). Tasc: Description of the tasc system used for muc-7. In *Seventh Message Understanding Conference*, MUC-7, Fairfax, Virginia.

Pedersen, T. (2010). Information content measures of semantic similarity perform better without sense-tagged text. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 329–332, Los Angeles, California. Association for Computational Linguistics.

Pedersen, T., Patwardhan, S., and Michelizzi, J. (2004). Wordnet::similarity: measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004*, HLT-NAACL–Demonstrations '04, pages 38–41, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ramshaw, L. and Marcus, M. (1995). Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*, pages 82–94, Cambridge, Massachusetts, USA.

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 1*, IJCAI'95, pages 448–453, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Richardson, M. and Domingos, P. (2006). Markov logic networks. *Machine Learning*, 62(1-2):107–136.

Riedel, S., Yao, L., and McCallum, A. (2010). Modeling relations and their mentions without labeled text. In *Proceedings of the 2010 European conference on Machine learning and knowledge discovery in databases: Part III*, ECML PKDD'10, pages 148–163, Berlin, Heidelberg. Springer-Verlag.

Riedl, M. and Biemann, C. (2012). Text segmentation with topic models. *Journal for Language Technology and Computational Linguistics*, 27(1):47–69.

Riloff, E. (1993). Automatically constructing a dictionary for information extraction tasks. In *Proceedings of the eleventh national conference on Artificial intelligence*, AAAI'93, pages 811–816. AAAI Press.

Riloff, E. (1996a). Automatically generating extraction patterns from untagged text. In *Proceedings of the thirteenth national conference on Artificial intelligence - Volume 2*, AAAI'96, pages 1044–1049. AAAI Press.

Riloff, E. (1996b). An empirical study of automated dictionary construction for information extraction in three domains. *Artifical intelligence*, 85(1-2):101–134.

Riloff, E. and Jones, R. (1999). Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of Artificial Intelligence*, AAAI '99/IAAI '99, pages 474–479, Menlo Park, CA, USA. American Association for Artificial Intelligence.

Rink, B. and Harabagiu, S. (2011). A generative model for unsupervised discovery of relations and argument classes from clinical texts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 519–528, Stroudsburg, PA, USA. Association for Computational Linguistics.

Rosario, B. and Hearst, M. A. (2004). Classifying semantic relations in bioscience texts. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Barcelona, Spain. Association for Computational Linguistics.

Rozenfeld, B. and Feldman, R. (2006a). High-performance unsupervised relation extraction from large corpora. In *Proceedings of the Sixth International Conference on Data Mining*, ICDM '06, pages 1032–1037, Washington, DC, USA. IEEE Computer Society.

Rozenfeld, B. and Feldman, R. (2006b). Ures: an unsupervised web relation extraction system. In *Proceedings of the COLING/ACL on Main conference poster sessions*, COLING-ACL '06, pages 667–674, Stroudsburg, PA, USA. Association for Computational Linguistics.

Rozenfeld, B. and Feldman, R. (2007). Clustering for unsupervised relation identification. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, CIKM '07, pages 411–418, New York, NY, USA. ACM.

Rozenfeld, B. and Feldman, R. (2008). Self-supervised relation extraction from the web. *Knowledge and Information Systems*, 17(1):17–33.

Sekine, S. (2001). Oak system (english sentence analyzer). `http://nlp.cs.nyu.edu/oak`.

Sekine, S. (2006). On-demand information extraction. In *Proceedings of the COLING/ACL on Main conference poster sessions*, COLING-ACL '06, pages 731–738, Stroudsburg, PA, USA. Association for Computational Linguistics.

Sekine, S., Sudo, K., and Nobata, C. (2002). Extended named entity hierarchy. In *LREC*.

Shinyama, Y. and Sekine, S. (2006). Preemptive information extraction using unrestricted relation discovery. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL '06, pages 304–311, Stroudsburg, PA, USA. Association for Computational Linguistics.

Snow, R., Jurafsky, D., and Ng, A. Y. (2005). Learning syntactic patterns for automatic hypernym discovery. In *Advances in Neural Information Processing Systems (NIPS 2005)*.

Soderland, S., Roof, B., Qin, B., Xu, S., Mausam, and Etzioni, O. (2010). Adapting open information extraction to domain-specific relations. *AI Magazine*, 31(3):93–102.

Stein, B., Sven, and Wißbrock, F. (2003). On cluster validity and the information need of users. In *The 3rd IASTED International Conference on Artificial Intelligence and Applications*, AIA'03, pages 404–413.

Steinbach, M., Karypis, G., and Kumar, V. (2000). A comparison of document clustering techniques. In *KDD Workshop on Text Mining*.

Stevenson, S., Fazly, A., and North, R. (2004). Statistical measures of the semi-productivity of light verb constructions. In *Proceedings of the Workshop on Multiword Expressions: Integrating Processing*, MWE '04, pages 1–8, Stroudsburg, PA, USA. Association for Computational Linguistics.

Stokes, N., Carthy, J., and Smeaton, A. F. (2004). Select: a lexical cohesion based news story segmentation system. *AI Communications*, 17(1):3–12.

Suchanek, F. M., Kasneci, G., and Weikum, G. (2007). Yago: A core of semantic knowledge. In *16th international World Wide Web conference*, WWW 2007, New York, NY, USA. ACM Press.

Sudo, K., Sekine, S., and Grishman, R. (2003). An improved extraction pattern representation model for automatic ie pattern acquisition. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 224–231, Sapporo, Japan. Association for Computational Linguistics.

Surdeanu, M., Turmo, J., and Ageno, A. (2006). A hybrid approach for the acquisition of information extraction patterns. In *In Proceedings of the EACL 2006 Workshop on Adaptive Text Extraction and Mining*, ATEM 2006. Association for Computational Linguistics.

Theodoridis, S. and Koutroumbas, K. (2009). *Pattern Recognition, Fourth Edition*. Academic Press, 4th edition.

Uzuner, O., Mailoa, J., Ryan, R., and Sibanda, T. (2010). Semantic relations for problem-oriented medical records. *Artifical intelligence in Medicine*, 50(2):63–73.

Uzuner, Ö., South, B. R., Shen, S., and DuVall, S. L. (2011). 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*, 18(5):552–556.

Van Dongen, S. (2000). *Graph Clustering by Flow Simulation*. PhD thesis, University of Utrecht.

Voorhees, E. M. and Buckland, L. P., editors (2009). *The Eighteenth Text REtrieval Conference Proceedings*, TREC 2009, Gaithersburg, Maryland. National Institute of Standards and Technology.

Voorhees, E. M. and Buckland, L. P., editors (2010). *The Nineteenth Text REtrieval Conference Proceedings*, TREC 2010, Gaithersburg, Maryland. National Institute of Standards and Technology.

Voorhees, E. M. and Buckland, L. P., editors (2011). *The Twentieth Text REtrieval Conference Proceedings*, TREC 2011, Gaithersburg, Maryland, November 15-18, 2011. National Institute of Standards and Technology.

Wang, W., Besançon, R., Ferret, O., and Grau, B. (2011). Filtering and clustering relations for unsupervised information extraction in open domain. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, pages 1405–1414, New York, NY, USA. ACM.

Wang, W., Besançon, R., Ferret, O., and Grau, B. (2012). Evaluation of unsupervised information extraction. In *Proceedings of the Eight International Conference on Language Resources and Evaluation*, LREC'12, Istanbul, Turkey. European Language Resources Association (ELRA).

Wang, W., Besançon, R., Ferret, O., and Grau, B. (2013). Regroupement sémantique de relations pour l'extraction d'information non supervisée. In *20ème Conférence sur le Traitement Automatique des Langues Naturelles*, TALN 2013, Les Sables-d'Olonne, France.

Wikipedia (2004). Wikipedia, the free encyclopedia. `http://www.wikipedia.org`.

Witten, I. H. and Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Wu, F. and Weld, D. S. (2007). Autonomously semantifying wikipedia. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, CIKM '07, pages 41–50, New York, NY, USA. ACM.

Wu, F. and Weld, D. S. (2010). Open information extraction using wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 118–127, Uppsala, Sweden. Association for Computational Linguistics.

Wu, Z. and Palmer, M. (1994). Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, ACL '94, pages 133–138, Las Cruces, New Mexico, USA. Association for Computational Linguistics.

Yan, Y., Okazaki, N., Matsuo, Y., Yang, Z., and Ishizuka, M. (2009). Unsupervised relation extraction by mining wikipedia texts using information from the web. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, ACL '09, pages 1021–1029, Suntec, Singapore. Association for Computational Linguistics.

Yangarber, R. and Grishman, R. (1998). Nyu: Description of the proteus/pet system as used for muc-7. In *In Proceedings of the Seventh Message Understanding Conference*, MUC-7.

Yangarber, R., Grishman, R., Tapanainen, P., and Huttunen, S. (2000). Automatic acquisition of domain knowledge for information extraction. In *Proceedings of the 18th conference on Computational linguistics - Volume 2*, COLING '00, pages 940–946, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yao, L., Haghighi, A., Riedel, S., and McCallum, A. (2011). Structured relation discovery using generative models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1456–1466, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yao, L., Riedel, S., and McCallum, A. (2012). Unsupervised relation discovery with sense disambiguation. In *Proceedings of The 50th Annual Meeting of the Association for Computational Linguistics*, pages 712–720. The Association for Computer Linguistics.

Yi, J. and Sundaresan, N. (1999). Mining the web for acronyms using the duality of patterns and relations. In *WIDM '99: Proceedings of the 2nd international workshop on Web information and data management*, pages 48–52, New York, NY, USA. ACM.

Zelenko, D., Aone, C., and Richardella, A. (2003). Kernel methods for relation extraction. *J. Mach. Learn. Res.*, 3:1083–1106.

Zhao, S. and Grishman, R. (2005). Extracting relations with integrated information using kernel methods. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 419–426, Ann Arbor, Michigan. Association for Computational Linguistics.

Zhu, J., Nie, Z., Liu, X., Zhang, B., and Wen, J.-R. (2009). Statsnowball: a statistical approach to extracting entity relationships. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 101–110, New York, NY, USA. ACM.