Knowledge engineering in the sourcing domain for the recommendation of providers

Molka Tounsi Dhouib

UNIVERSITÉ CÔTE D'AZUR

ÉCOLE DOCTORALE
SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

THÈSE DE DOCTORAT

Ingénierie des connaissances dans le domaine du sourcing pour la recommandation de prestataires

Molka Tounsi

Laboratoire d'Informatique, Signaux et Systèmes de Sophia Antipolis

Présentée en vue de l'obtention du grade de docteur en Informatique

Dirigée par : Mr. Andrea Tettamanzi, Professeur, Université Côte d'Azur

Dirigée par : Mme. Catherine FARON ZUCKER, Maître de conférences, HDR, Université Côte d'Azur

Soutenue le : 26 Mars 2021

Devant le jury, composé de :

Président du jury :

Mr. Fabien Gandon, Directeur de Recherche, INRIA

Rapporteurs :

Mme. Nathalie Pernelle, Professeure, Université Sorbonne Paris Nord

Mme. Marie-Christine Rousset, Professeure, Université de Grenoble Alpes

Examinateur :

Mme. Sylvie Despres Professeure, Université Sorbonne Paris Nord

Mr. Petko Valtchev, Professeur Adjoint, Université de Montréal

Invité :

Mr. Nicolas Bridey, Co-fondateur, Silex

silex

Knowledge engineering in the sourcing domain for the recommendation of providers

Jury

Président du jury
Mr. Fabien Gandon, Directeur de Recherche, INRIA

Rapporteurs
Mme. Nathalie Pernelle, Professeure, Université Sorbonne Paris Nord
Mme. Marie-Christine Rousset, Professeure, Université de Grenoble Alpes

Examinateur
Mme. Sylvie Despres, Professeure, Université Sorbonne Paris Nord
Mr. Petko Valtchev, Professeur Adjoint, Université de Montréal

Invité
Mr. Nicolas Bridey, Co-fondateur, Silex

Directeurs de thèse
Mr Andrea Tettamanzi, Professeur, Université Côte d'Azur
Mme. Catherine FARON ZUCKER, Maître de conférences, Université Côte d'Azur

Résumé

Cette thèse de doctorat CIFRE s'inscrit dans le cadre d'un projet de recherche collaboratif entre le laboratoire I3S de l'Université Côte d'Azur et la société Silex et aborde le domaine des systèmes de recommandation. De nos jours, les systèmes de recommandation sont devenus populaires et se sont répandus dans de nombreux domaines d'application (recommandation de films, actualités, restaurants, produits, services financiers, etc). Ils fournissent des suggestions aux utilisateurs pour les aider dans leur recherche d'information, en essayant de répondre au mieux à leurs besoins. Silex est une start-up qui développe un outil de sourcing Software-as-a-Service permettant aux entreprises de fournir une description de leurs activités professionnelles, de leurs offres et/ou des services qu'elles recherchent en langue naturelle (actuellement le français). Dans ce contexte, l'objectif de cette thèse est de proposer un système d'aide à la décision en exploitant les connaissances sémantiques extraites à partir des descriptions textuelles des demandes de prestation et des prestataires, afin de recommander des prestataires pertinents pour une demande de prestation.

Les contributions de cette thèse sont les suivantes. Premièrement, nous avons proposé un vocabulaire pour le domaine du sourcing afin d'annoter sémantiquement les descriptions textuelles des prestataires et des demandes de prestation. Ce vocabulaire a été construit en réutilisant et en intégrant des vocabulaires existants. Deuxièmement, nous avons proposé une méthode d'alignement automatique afin d'établir la correspondance entre différents concepts des vocabulaires considérés. Cette approche se base sur des règles exploitant l'espace des plongements lexicaux et des mesures sur des groupes d'étiquettes pour découvrir les relations entre concepts. Troisièmement, nous avons proposé un algorithme d'extraction des entités nommées à partir des descriptions textuelles des demandes de prestation et des prestataires et un algorithme d'annotation sémantique de ces descriptions, basé sur le liage des entités extraites avec les concepts du vocabulaire défini. Quatrièmement, nous avons proposé un algorithme de recommandation de prestataires qui exploite ces annotations sémantiques. Finalement, nous avons étudié l'apport de l'utilisation de connaissances ontologiques afin d'améliorer notre système d'aide à décision pour le domaine du sourcing.

Mots-clefs: Système de recommandation, ontologie, alignement d'ontologies, sourcing.

Abstract

This CIFRE doctoral thesis is part of a collaborative research project between the I3S laboratory of the University of Côte d'Azur and the Silex company, and addresses the field of recommendation systems. Nowadays, recommendation systems have become popular and widespread in many application domains (recommendation of movies, news, restaurants, products, financial services, etc.). They provide users with suggestions to help them in their search for information, trying to best meet their needs. Silex is a start-up that develops a Software-as-a-Service sourcing tool that allows companies to provide a description of their professional activities, their offers and/or the services they are looking for in natural language (currently French). In this context, the objective of this thesis is to propose a decision support system by exploiting the semantic knowledge that are extracted from the textual descriptions of requests for services and providers, in order to recommend relevant providers for a service request.

The contributions of this thesis are the following. First, we proposed a vocabulary for the sourcing field in order to semantically annotate the textual descriptions of providers and requests for services. This vocabulary was built by reusing and integrating existing vocabularies. Second, we proposed an automatic alignment method to establish the correspondence between different concepts of the considered vocabularies. This approach is based on rules exploiting embedding space and measurements on groups of labels to discover the relationships between concepts. Third, we proposed an algorithm for extracting named entities from the textual descriptions of service requests and providers, and an algorithm for semantic annotation of these descriptions, based on the linking of the extracted entities with the concepts of the defined vocabulary. Fourth, we proposed a provider recommendation algorithm that exploits these knowledge extracted. Finally, we studied the contribution of using ontological knowledge to improve our decision support system for the sourcing domain.

Keywords: Recommender system, ontology, ontology alignment, sourcing domain.

To your soul my dear daddy,
You never left my mind,
I miss you ...

To Anas, Ahmed and Tasnime

Remerciements

Merci à tous ceux que j'ai rencontrés pendant toutes ces années et qui ont pu m'aider à arriver à ce moment.

Je voudrais d'abord exprimer ma profonde reconnaissance à mes directeurs de thèse Catherine Faron Zucker et Andrea Tettamanzi pour leur confiance, leur encadrement enrichissant et les conseils qui m'ont permis d'avancer tout le long de ma thèse. L'attention et le temps qu'ils m'ont accordés, leur soutien et leurs encouragements m'ont permis de surmonter les moments difficiles. Catherine et Andrea, c'est une chance et un honneur d'avoir été une de vos doctorants.

Je ne remercierai jamais assez Catherine Faron Zucker de m'avoir initiée au monde de la recherche scientifique, et de m'avoir encadrée depuis mon premier stage de recherche et de m'accompanger encore pour l'après thèse.

Ma profonde reconnaissance à Fabien Gandon d'avoir accepté la présidence du jury de cette thèse et aussi pour tous les conseils et les opportunités qu'il m'a accordés tout au long de mon parcours chez Wimmics.

Je tiens à remercier aussi Nathalie Pernelle professeure à l'université Sorbonne Paris Nord et Marie-Christine Rousset professeure à l'université de Grenoble Alpes d'avoir accepté de rapporter cette thèse. Je remercie également Sylvie Despress professeure à l'université Sorbonne Paris Nord et Petko Valtchev professeur adjoint à l'université de Montréal d'avoir accepté de faire partie du jury de ma soutenance.

J'adresse aussi mes remerciements à Nicolas Bridey et Quentin Fournela, fondateurs de Silex pour m'avoir offert l'opportunité de travailler au sein de Silex. Merci pour la confiance et la liberté qu'ils m'ont accordées durant ces années, ce qui a fortement contribué aux résultats présentés dans ce manuscrit.

Merci aux membres passés et présents de l'équipe WIMMICS que j'ai côtoyés. Les échanges enrichissants ont permis une ambiance de travail conviviale et m'ont permis de grandir et de progresser. Je tiens à saluer tout particulièrement Christine Foggia pour son support pour toutes mes missions.

Merci également aux membres passés et présents de l'équipe Silex et en particulier l'équipe technique: Jerôme pour son soutien et son écoute, Hasnaa pour sa contribution à la partie de l'extraction des connaissances dans le cadre de son stage, sans oublier Camille, Kévin, Hervé, Antoine et Thibault.

Je remercie aussi toute l'équipe produit pour les échanges que nous avons eus, et surtout pour leur travail d'annotation manuelle qui était précieux pour la validation de mes travaux.

Merci à ma chère famille et mes proches de me soutenir, de m'encourager et m'entourer d'amour depuis longtemps. J'ai une pensée particulière à ma mère pour son soutien, son amour, et son appui moral malgré la distance qui me sépare d'elle. Merci à mon frère, ma sœur et leurs petites familles pour leur amour, et ma belle-famille qui m'a beaucoup encouragée, et particulièrement ma belle-mère qui n'a pas hésité à faire des sacrifices pour me soutenir pendant ces années.

Enfin, je ne pourrai jamais exprimer assez de gratitude à ceux qui m'apportent l'amour, la tendresse et le réconfort dont j'ai besoin pour avancer. Je remercie mon bien-aimé Anas d'avoir toujours cru en moi, de toujours faire de son mieux pour me soutenir, m'encourager et me donner la force de surmonter les difficultés rencontrées pendant ma thèse. Merci d'être si patient, si arrangeant et d'avoir toujours su être là. Je remercie mes petits trésors, Ahmed et Tasnime, d'être toujours ma source de bonheur, d'amour et de motivation. Tout seul, on va plus vite, ensemble on va plus loin.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents

1.1 Context

Nowadays, companies are collecting masses of data through multiple channels with multiple formats, and facing big challenges to analyze them and create value. To deal with that, companies are accelerating the integration of digital technologies, especially Artificial Intelligence (AI) technology that is revolutionizing their processes.

The sourcing domain is one of the most promising domains for AI. Used in the procurement, sourcing is the action of searching, identifying and evaluating an *ad hoc* supplier, in order to meet an identified need (for goods or services) formulated by a company or a service or a department of that company. The evaluation of providers is based on multiple criteria, such as cost, deadline, innovation, quality, proximity, response capacity on needs, and production capacity. Traditional sourcing methods are based on using online paid databases or specialized databases in a specific domain, and on trade shows, federations, and unions. Unfortunately, with today's fast-paced and ever-changing environment, manual sourcing procedures often lead to a lack of transparency, incorrect supplier lists, and inconsistent purchasing management. These drawbacks involve big losses for both buyers and suppliers.

The *Silex* company[1] develops an e-sourcing tool with the ultimate goal of providing a framework to steer and optimize an lookup and collection of supplier offers following a model facilitating their analysis. Using this framework, the purchasers will collect information on providers, their products or services and prices. Based on this information and the purchasers' requirements, reliable comparisons will be made to suggest a list of potential providers. Automatizing the e-sourcing process has multiple benefits: on the one hand, the purchasing department has a better follow-up, real time savings on order and delivery times, and more efficient and transparent purchasing; on the other hand, the provider could boost their activities, improve collaboration, increase their revenue, and increase transparent exchange.

This *CIFRE* thesis takes place in the context of a collaborative research project between the *Silex* company and the WIMMICS research team[2] from Inria and I3S at Université Côte d'Azur, aiming to develop IA methods and tools to be integrated within the *Silex* sourcing platform. This platform supports two main user communities: (i) service requestors and (ii) providers. Users provide a textual description of their professional activities, their offers and/or services in the French language. The platform automatically analyzes these textual descriptions in order to better and faster evaluate opportunities, with more targeted sourcing.

The goal of this thesis is to design a recommender system (RS) relying

[1] https://www.silex-france.com/silex/
[2] https://team.inria.fr/wimmics/

on the prediction of relevant providers that are likely to be of interest for a service request based on contextual analysis of these data. Two main challenges have been identified:

1. How can we analyze expressions in natural language (national localized context specific to each country)?

2. How can we model these expressions in a formal representation allowing to reason about them?

1.2 Our research focus

In this thesis, we focus on the introduction of semantics into the *Silex* platform by conceptualizing the knowledge involved in sourcing, in order to be able to automatically reason on service requests and providers descriptions, and improve the recommender process.

1.2.1 Domain knowledge modeling for sourcing data representation

A first ontological engineering work consists of the design of a semantic repository to describe the sourcing data. This involves exploiting heterogeneous sources such as classification or metadata repositories to build it. There are two main challenges in this first part:

1. Define a precise and complete domain knowledge that describes the sourcing domain, in phase with the *Silex* data.

2. Integrate the multilingual dimension of the data sources used into the domain knowledge to position the service request or provider in both French and European markets.

1.2.2 Design and adaptation of algorithms for knowledge extraction

This involves the implementation of Natural Language Processing (NLP) algorithms to identify the relevant key phrases to be extracted from the textual descriptions of providers and service requests. Our challenges in this part are:

1. Deal with short descriptions and distinguish between the real need and the general context of the service request.

2. NLP algorithms have to be able to process texts in different languages.

1.2.3 Recommend providers based on the formal representation of descriptions

The aim is to define a recommender approach based on the matching of formal representations of providers and service requests, by exploiting the semantic proximity captured by the designed domain knowledge (for instance, a company with a hiring need will be interested in service providers presenting themselves as recruiters, but perhaps also in others presenting themselves as headhunters, specialized in the recruitment of highly qualified people). The provider recommender then involves developing evaluation metrics between an expressed need and the matching providers.

1.3 Our methodology

In our recommender scenario, when a new service request is published on the *Silex* platform, a set of the most relevant providers for it should be automatically suggested, either directly to the user or to the *Silex* sale's staff for validation.

Figure 1.1 shows the proposed overall workflow. It comprises five steps: (i) construction of a vocabulary for the sourcing domain, (ii) entity recognition from the textual descriptions of service requests and providers, (iii) entity management, (iv) vector representation of service requests and providers, and (v) recommender algorithm.

In the following, we summarize the methodology we developed to achieve this goal.

1.3.1 Building a vocabulary for the sourcing domain

The preliminary step of our approach is the construction of a vocabulary to capture the sourcing knowledge. Our proposal relies on the use of domain knowledge that can be captured into a thesaurus or an ontology. In the following, we will refer to vocabulary indistinctly as a thesaurus or an ontology. The aim is to semantically annotate the textual descriptions of companies and service requests with five types of knowledge: (i) skills, (ii) occupations, (iii) products (i.e. goods and services), and (v) business activities. We built a modular sourcing vocabulary by identifying and combining several relevant standard metadata repositories.

1.3.2 Proposal of an ontology alignment approach

The second step is the proposal of an alignment method and its development to automatically align the above-mentioned vocabularies, which where partially overlapping, in order to get a complete vocabulary.

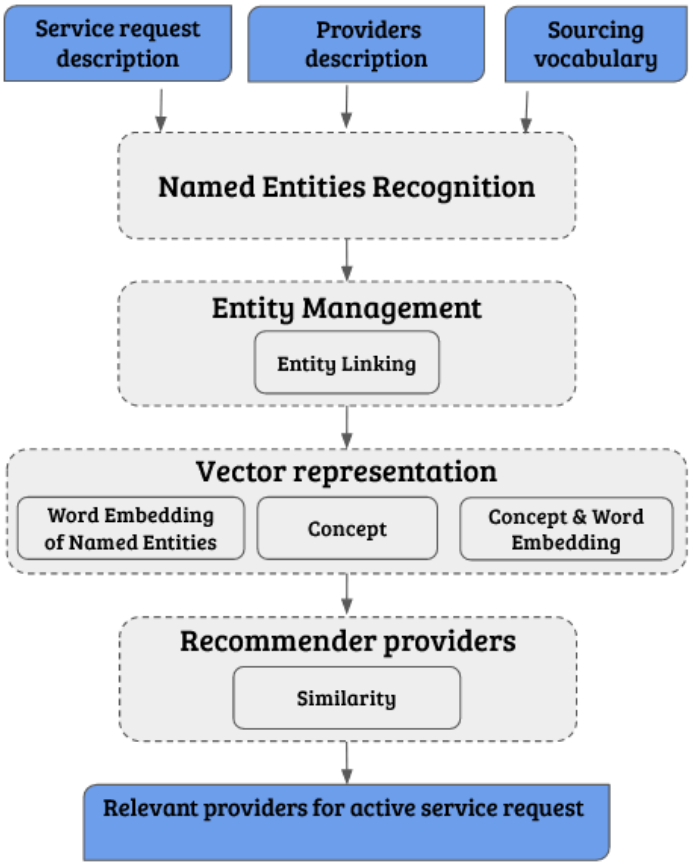


Figure 1.1: Overview of our recommendation approach.

1.3.3 Proposal of an approach for sourcing named entity recognition and linking

The third step is the definition of an NLP algorithm to extract and link the relevant knowledge pieces, i.e., occupations, skills, products, and business activities, from the textual descriptions, i.e., service requests and company descriptions.

1.3.4 Proposal of a recommender algorithm

The final step is to propose a formal representation of each service request or provider that summarizes the semantics of the entities extracted from their descriptions and define metrics to measure the similarity between these representations used to make the matching. This similarity measure is the backbone of our recommendation algorithm.

1.4 Our contributions

In this thesis, our main contributions are as follows:

1. The construction of a sourcing domain vocabulary by using external and internal metadata repository. The results of our work have been published in the proceedings of French Knowledge Engineering Days (IC) [Dhouib et al., 2018].
2. The proposal of a new ontology alignment approach using embedding and radius measure. The results of our work have been published in the proceedings of the International Conference on Semantic Systems [Dhouib et al., 2019].
3. The proposal of a new named entity recognition algorithm combining several types of extracted features describing the textual content such as: (i) semantics, (ii) syntax, (iii) word characters, and (iv) position of words. The results of our work have been published in the proceedings of the National Conference on Practical Applications of Artificial Intelligence (APIA) [Daoud et al., 2020].
4. The proposal of a recommender system approach based on domain knowledge and similarity between the vector representations. The results of our work have been published in the proceedings of the Web Intelligence conference [Dhouib et al., 2020].

1.5 Thesis outline

This thesis is organized into the following chapters:

- Chapter 2 defines the relevant notions used in this thesis.
- Chapter 3 presents the modular vocabulary we developed for the sourcing domain.
- Chapter 4 describes the ontology alignment approach we developed to match the different ontologies we reused. It also presents the evaluation of this approach conducted several other ontologies.
- Chapter 5 presents our knowledge extraction approach by defining a named entity recognition approach and an entity linking method.
- Chapter 6 describes the approach we developed to match the companies community and the service request community and the recommender algorithm based on it.
- Chapter 7 summarizes the results of our thesis and discusses future research directions.

Chapter 2

Background knowledge

Contents

2.1 Introduction

In this chapter, we introduce some background knowledge that will be used through out the thesis. Section 2.2 presents the different notions and languages related to the semantic Web domain. Sections 2.3 and 2.4 introduce respectively the knowledge extraction and the recommender systems domains. Section 2.5 reports a brief overview of the state of the art of vector representations relevant to textual data. Section 2.6 gives a brief introduction on the machine and deep learning algorithms used in the named entity recognition task. Section 2.7 defines the different evaluation metrics used throughout our work. Finally, Section 2.8 concludes this chapter.

2.2 Semantic Web

In 1999, Tim Berners-Lee presented for the first time his vision of the Semantic Web. *"The Semantic Web is not a separate Web, but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in co-operation"* [Berners-Lee et al., 2001]. Tim Berners-Lee's goal was to extend the current Web with metadata by allowing both machines and humans to better manipulate information and make meaningful interpretations. Therefore, we are no longer talking only about a Web of documents but a Web of data [Bizer et al., 2011].

2.2.1 Knowledge organization systems

With the emergence of the semantic Web, the notion of ontology has experienced a new rise. Formerly reserved for the field of philosophy, and later on that of knowledge representation, ontologies now represent the backbone of the semantic Web technology. In the literature, several definitions of ontology have been presented and evolved over the time. The most quoted one is given by Gruber [Gruber, 1995] *"an ontology is a formal, explicit specification of a shared conceptualization"*. Typically, an ontology is a set of concepts or classes of objects that share common characteristics and relationships or properties [Antoniou and Van Harmelen, 2004]. In addition, an ontology can include a set of axioms which ascertain the coherence of the model, such as disjointness, equivalence, cardinality axioms for concepts, transivity, and functional or inverse axioms for properties.

There are several related conceptual structures to capture knowledge [Zacklad, 2007]:

- A thesaurus is set of terms of a specific domain. These terms are enriched by semantic relations such as hierarchical relation, equivalence relation and association relation.

- A taxonomy is a list of terms of a domain organized in a hierarchical way.

- The term of knowledge graph has been presented by Google as a new trend in 2012. A knowledge graph is usually interpreted as a collection of interlinked entities and relations between those entities. So it may used as a synonym for ontology. In some contexts, it is used to refer to any knowledge base that is represented as a graph [Krötzsch, 2017; Ehrlinger and Wöß, 2016].

Based on the definitions of the semantic Web and ontologies, we can highlight two main benefits of these technologies: (i) Interoperability, which reposes on sharing and exchanging data across Web applications and agents; (ii) Inferencing, which means the ability of the system to derive new knowledge and new facts.

2.2.2 Semantic Web languages

2.2.2.1 Resource description framework

Resource Description Framework (RDF)[1] is a graph model for representing and structuring data and metadata. RDF is based on the notion of triple (subject, predicate, object), where the subject represents the resource to be described, the predicate represents the property, and the object represents a literal value or another resource. Triples can also be seen as the arcs of a labeled oriented graph that would be distributed on the web. We take for example the following sentence *"Silex provides a sourcing platform"*. This sentence describes the company *Silex* by the fact that it provides a sourcing platform. This sentence can be represented in RDF by the triple `(Silex, provides, Sourcing_platform)`. RDF statements can be represented in a variety of syntaxes, among which RDF/XML,[2] Terse RDF Triple Language (Turtle),[3] and JSON-based serialization (JSON-LD).[4]

2.2.2.2 Resource description framework schema

Resource Description Framework Schema (RDFS) [5] is a language to represent the vocabulary used to describe properties and classes used in RDF knowledge graphs. In our example, the `provides` property and the class `Company` are part of the *Silex* RDFS vocabulary.

[1] https://www.w3.org/RDF/
[2] https://www.w3.org/TR/rdf-syntax-grammar/
[3] https://www.w3.org/TR/turtle/
[4] https://www.w3.org/2018/jsonld-cg-reports/json-ld/
[5] https://www.w3.org/TR/2000/CR-rdf-schema-20000327/

2.2.2.3 Ontology web language

Ontology Web Language (OWL)[6] is a W3C recommendation to add more vocabulary for the description of properties and classes, which allow supplementary inference capabilities. We can express for example restrictions on the value of properties or their cardinality, or algebraic properties (symmetrical, transitive, functional, inverse property, disjointness).

2.2.2.4 Simple knowledge organization system

Simple Knowledge Organization System (SKOS)[7] is a W3C recommendation based on RDF and OWL dedicated to represent terminological resources, thematic classifications, glossaries, thesauri, or any other type of controlled and structured vocabulary. SKOS provides several primitives to declare preferred labels or synonymous labels for each concept in each language and synonymy or hyponymy relations [Gandon, 2008].

2.2.2.5 SPARQL protocol and RDF query language

SPARQL Protocol and RDF Query Language (SPARQL)[8] provides a query language to manipulate RDF graphs.

2.2.3 Ontology alignment

Ontology alignment is the process of discovering correspondences between concepts and relations from different ontologies. It represents the key ingredient for the semantic interoperability and to solve the semantic heterogeneity problem.

We adopt the ontology alignment definition introduced by [Euzenat et al., 2007; Shvaiko and Euzenat, 2011]. A correspondence between a source ontology O_1 and a target ontology O_2 is defined as a tuple $\{(e_1, e_2, r, con)\}$, where:

- e_1 is an entity in O_1,
- e_2 is an entity in O_2,
- r is the semantic relationship between e_1 and e_2 such as equivalence($\equiv$), more general ($\sqsupseteq$), and
- *con* is the confidence score (typically in the [0, 1] range) holding for the correspondence between e_1 and e_2.

[6] https://www.w3.org/OWL/
[7] https://www.w3.org/2004/02/skos/
[8] https://www.w3.org/TR/rdf-sparql-query/

Generally, the most commonly used semantic relations are equivalence and subsumption relations. For OWL ontologies we can use *owl:equivalentClass* for equivalent alignment of classes, *owl:sameAs* for equivalence alignment of individuals and *rdfs:subClassOf* for subsumption alignment of classes. For SKOS vocabularies, we can use *skos:narrowMatch* and *skos:broadMatch* for hyponymy relations between concepts, and *skos:exactMatch* or *skos:closeMatch* for synonymy relations. Alignments can be of various cardinalities: (i) one-to-one (1:1), (ii) one-to-many (1:m), (iii) many-to-one (n:1), or (iv) many-to-many (n:m). There are two kinds of matches: (i) a simple match is about linking two atomic entities represented by their identifiers; and (ii) a complex match allows to express logical formulas between entities [Thiéblin et al., 2017].

2.3 Information and knwledge extraction

Information extraction (IE) [Wimalasuriya and Dou, 2010; Singh, 2018] is the task of automatically extracting relevant knowledge from unstructured and/or semi-structured data, and converting that into a representation suitable for machine processing.

Similar to IE, knowledge extraction (KE) is the task of extracting knowledge from structured and unstructured data. The main difference between KE and IE is in the case of IE the extraction result is converted in to a relational schema, whereas for KE the result of extraction requires the reuse of existing formal knowledge or the generation of a schema based on source data.

2.3.1 Natural language processing

Natural Language Processing (NLP) [Assal et al., 2011] is a scientific and engineering field concerned with studying the structure and rules of languages to allow computers to analyse and process in order to derive meaning from text and speech.

2.3.2 Sequence labeling

Sequence labeling [Jagannatha and Yu, 2016] is a NLP task which involves of assigning a class or label to each token in a given input sequence. It includes named entity recognition (NER), syntactic chunking and part of speech (POS) tagging.

2.3.3 Named entity recognition

Named entity recognition (NER) [Grishman and Sundheim, 1996; Nadeau and Sekine, 2007] also called entity identification or entity extraction, is

probably the first step towards knowledge extraction from text. NER seeks to locate entities mentioned in unstructured text into pre-defined categories such as person names, organizations, locations, time expressions, quantities, etc. It is generally composed of two main phases: (i) identify named entities, and (i) classify entities into predefined categories.

2.3.4 Named entity linking

Named Entity linking (NEL)[Rao et al., 2013] is the task of linking an entity mention that has been identified in a text with its corresponding entities in a knowledge base such as Wikidata,[9] DBpedia,[10] YAGO[11] or any other knowledge graph.

2.4 Information filtering

Information filtering (IF) [Hanani et al., 2001] is about removing redundant or unwanted information from large information flows, and managing the information overload in order to expose to users only information that is relevant to them. Recommender systems (RS) is an active information filtering systems. The first recommender system definition was given by [Resnick and Varian, 1997]: *"people provide recommendations as inputs, which the system then aggregates and directs to appropriate recipients"*. Later on, [Burke, 2002] refines this definition and introduces RS as *"any system that produces individualized recommendations as output or has the effect of guiding the user in a personalized way to interesting or useful objects in a large space of possible options"*. So, we can define RS as a system that helps users to answer their need and provide them with a personalized suggestion in order to pick the most relevant elements. RS are composed of two main entities: (i) the user who plays two roles at the same time, providing opinion about items and receiving the recommendation, and (ii) items such as products, services, articles, movies, music, or social connections, etc.

Some examples of e-commercial RS are Amazon[12] and Netflix.[13] In 1998, Amazon introduced its item-to-item collaborative filtering algorithm [Linden et al., 2003]. MacKenzie et al. [2013] estimated that 35% of consumer purchases on Amazon come from product recommendations. In 2006, Netflix created the Netflix award competition to improve the accuracy of its film recommendation system by 10%. Netflix mentioned that 70% of what users watch is the result of personalized recommendation [MacKenzie et al., 2013].

[9] https://www.wikidata.org/wiki/Wikidata:Main_Page
[10] https://wiki.dbpedia.org/
[11] https://yago-knowledge.org/
[12] https://www.amazon.fr/
[13] https://www.netflix.com/fr/

Many challenges are facing the development of RS [Singh et al.; Shah et al., 2016]:

- Cold start: means that the information about items or users are not yet sufficient to provide the best results. This problem is usually associated with the lack of valuable user interactions when a new item or new user is added to the system.
- Sparsity: is also related to the lack of information because the user rated just some items, but no interaction is available on other items. So if only few items are evaluated, it will be difficult to determine his/her taste.
- Popularity bias: means that the system fails to recommend items that are not popular.
- Over-specialization: is the fact that the system recommends only the items rated by users and ignores all the items that are different from anything that user has seen before. This can be a problem if the user wants to try something new.
- Scalability: relies on the complexity of the RS and the ability of a system to react with a large dataset.

2.5 Vector representations of textual data

2.5.1 Bag of words

A Bag of words (BOW) represents a text as a list of tokens based on either the occurrence or frequency of each word in the textual document. The dimension of the feature space is the number of all different words in all documents [Feldman and Sanger, 2007]. The limitation of this model is that it ignores the semantic relations between words.

2.5.2 Bag of concepts

A Bag of concepts (BOC) represents the text as a list of concepts and not a list of tokens. [Sahlgren and Cöster, 2004] introduce this representation by assuming that the meaning of a text can be approached by the union of keyword concepts in the text. Thus, a text is represented by the weighted vector sum of concept vectors corresponding to the terms present in the document [Täckström, 2005].

2.5.3 Word embedding

Word embedding is a distributed word representation that leverages the semantics of words by mapping them to vectors of real numbers, where each

dimension of the embedding represents a latent feature of the word [Turian et al., 2010; Gromann and Declerck, 2018]. Word embedding models are trained through a shallow neural network architecture to embed words in a dense continuous vector space, based on their linguistic contexts in a corpus, to preserve the semantic and syntactic similarities between words. As a result, words appearing in similar contexts in a text are represented by similar vectors.

Many models produce word embeddings: (i) Word2vec [Mikolov et al., 2015] provides two neural architectures, namely Skip-Gram and CBOW. Skip-Gram takes as input a word and tries to predict its context, whereas CBOW receives as input the context of a word (i.e. the words around it in a sentence) and tries to predict the word in question, (ii) Glove [Pennington et al., 2014] is a count-based model using the global matrix factorization to calculate the co-occurrence of words in the corpus, (iii) fastText [Bojanowski et al., 2017] is a library for learning word representations that published a pre-trained word vectors for several languages trained on Wikipedia. Unlike the other models, fastText provides a vector representation for each character n-gram [Gromann and Declerck, 2018].

The state of the art of word embedding models has recently evolved to what is known as contextual embedding. Embedding from Language Models (ELMo) [Peters et al., 2018a] is a deep contextualized word representation using a Bi-directional LSTM architectures able to create contextual word embeddings. The ELMo idea is to not use a fixed embedding for each word, but to look at the entire sentence before generating the embedding. As a result, the same word can have multiple representations based on its context. Bidirectional Encoder Representation from Transformers (BERT) [Devlin et al., 2018] relies on an architecture that uses attention layers to better capture the semantic relations inside the embedding. BERT is pre-trained on a large corpus of unlabelled texts composed from the entire Wikipedia and book corpus. Bert has two main pre-training objectives, the prediction of "hidden words" and the next sentence of a sequence. CamemBERT [Martin et al., 2019] is based on the RoBERTa [Liu et al., 2019] architecture which is a robust and optimized approach from BERT. The difference between CamemBERT and BERT can be summarized in these three points: (i) CamemBERT has only one pre-training objective which is the prediction of "hidden words"; (ii) it is pre-trained on 138 GB of French text; and (ii) it uses new hyper-parameters. In fact, it chooses the words to be predicted dynamically by randomly masking certain words in a sequence.

2.5.4 Graph embedding

Inspired by word embedding, graph embedding methods consist in learning a continuous vector space for each entity (node/or edge) of a graph. As a result, similar entities have similar vector representations. Several graph

embedding methods are presented in the literature. Node2vec [Grover and Leskovec, 2016] is an algorithmic framework that aims to create embedding for nodes in a graph. Node2vec is based on random walks which means that the algorithm starts a walk at a random node and performs a series of steps where each step goes to a random neighbor. Translating Embedding for Modeling Multi-relational data (TransE) [Bordes et al., 2013] provides a very simple and efficient way to capture the structure of a knowledge graph. It generates embedding entities based on the sum of the source entity vector and the translation vector of relation in such a way that they are as close as possible to the target vector. However, this model is not able to represent all other relations that do not link only two entities (i.e. 1-N, N-1 and N-N relations). To overcome this limitation, many models have been developed to extend TransE. TransH [Wang et al., 2014] presents a more flexible model by interpreting a relation as a translating operation on a hyperplane. TransR [Lin et al.] is a generalization of TransH. The basic idea of TransR is to use two distinct spaces, one for entities and the other for multiple relations. TransG [Xiao et al., 2015], produces multiple vectors for a relation to take into account the multiple meanings of this relation. pTransE [Lin et al., 2015] takes into account the relation paths between entities by the summing of all relations in a path.

2.6 Machine learning and deep learning algorithms

2.6.1 Hidden markov model

The hidden Markov model (HMM) is a statistical model used in sequence labeling problem. HMM analyzes the sequence pattern of observed symbols in order to interpret the non observable process. An HMM consists of a double stochastic process, in which the hidden stochastic process can be indirectly inferred by analyzing the sequence of observed symbols of another set of stochastic processes [Awad and Khanna, 2015; Nguyen and Guo, 2007].

2.6.2 Maximum entropy markov model

Maximum entropy Markov model (MEMM) [McCallum et al., 2000] is a discriminative model that combines HMM and maximum entropy. The basic idea of MEMM relies on the fact that the unknown values to be learnt are connected in a Markov chain rather than being conditionally independent of each other.

2.6.3 Support vector machine

Support Vector Machine (SVM) [Cortes and Vapnik, 1995] is a supervised learning algorithm which transforms training data into higher dimensions,

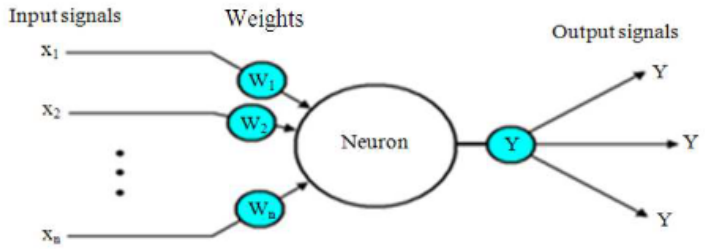


Figure 2.1: A simple NN architecture. Source: [Mohammed and Omar, 2012]

and searches for a linear optimal separating hyperplane.

2.6.4 Bayesian network

Bayesian Network (BN) is a probabilistic graphical model that represents knowledge about an uncertain domain. Each node in the directed acyclic graph corresponds to a random variable, and each edge represents the conditional probability for the corresponding random variables [Yang, 2019].

2.6.5 Neural network

A Neural Network (NN) is a mathematical model inspired by the human brain to process information and it is composed by many layers. The perceptron in Figure 2.1 is the simplest kind of NN, which has only two layers (the input nodes receive the feature values and the output nodes produce the NER result), and the link weights represent dependence relations [Mohammed and Omar, 2012].

2.6.6 Convolutional network

A Convolutional Network (CNN) is a regularized version of a multilayer perceptron or fully connected network. This regularization is based on two types of layers: (i) a convolutional layer based on a convolutional filtering, and (ii) a pooling layer that reduces the spatial size in order to limit the amount of neurons and the complexity of the minimization problem [Cagli et al., 2017].

2.6.7 Recurrent neural networks

Recurrent Neural Networks (RNN) [Gers et al., 1999] are a family of neural networks operating on sequential data. In contrast to traditional neural networks which assume that all inputs and outputs are independent from each other, RNN performs the same task for every element of a sequence, and the output depends on the previous computations.

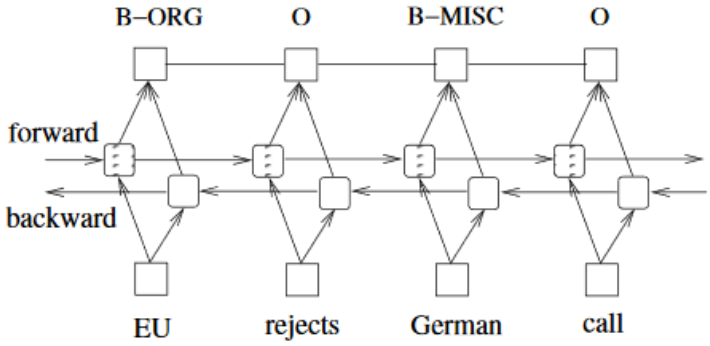


Figure 2.2: A Bi-LSTM-CRF-architecture. Source: [Huang et al., 2015]

2.6.8 Long short-term memory networks

The main problem of RNN is that they are looking back only a few steps. Long Short-term Memory Networks (LSTM), a particular type of RNN, are designed to avoid this problem through the use of a memory-cell to capture long-term dependencies [Gers et al., 1999]. The goal of LSTM is to learn information from context. The main characteristics of LSTM are: (i) the ability to operate on sequential data, and (ii) the ability to capture long term dependencies thanks to a memory-cell.

2.6.9 Bidirectional long a short-term memory-conditional - random fields networks

Bidirectional Long Short-Term Memory-Conditional Random Fields network (Bi-LSTM-CRF) as shown in figure 2.2 is obtained from the combination of a Bi-LSTM model and a CRF classifier. Bi-LSTM model [Graves and Schmidhuber, 2005] is composed of two LSTMs, one that processes the input sequence from left to right, and the other that processes the input in the reverse direction (i.e., from right to left). CRF [Lafferty et al., 2001] is an undirected graphical model and is partially similar to HMMs. The difference between CRF and HMM is that HMM simply works on the word type of tokens, while CRF works on a set of features defined automatically from input tokens during the training process [Sutton et al., 2012; Poostchi et al., 2018].

2.7 Evaluation measures

General evaluation metrics are used to analyse the performance of a model for different tasks (e.g. ontology alignment, NER, RS). These metrics are based on the followings: (i) True Positive (TP) is the number of positive

instances correctly assigned; (ii) False Positive (FP) is the number of positive instances incorrectly assigned; (iii) False Negative is the number of negative instances incorrectly assigned; and (iv) True Negative (TN) is the number of negative instances correctly assigned.

2.7.1 Precision

Precision is used to check the degree of correctness of the model.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{2.1}$$

2.7.2 Recall

Recall is used to check the degree of completeness of the model.

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}}. \tag{2.2}$$

2.7.3 F1-measure

F1-measure is the harmonic average of recall and precision.

$$\text{F-measure} = 2 \cdot \frac{\text{recall} \cdot \text{precision}}{\text{recall} + \text{precision}}. \tag{2.3}$$

The precision, recall and F1-measure are commonly used to evaluate NER [Konkol and Konopík, 2013; Jiang et al., 2016a; Tsai et al., 2006], ontology alignment [Euzenat, 2007; Ochieng and Kyanda, 2018], and RS [Lerato et al., 2015; Gunawardana and Shani, 2009; Del Olmo and Gaudioso, 2008].

2.7.4 Precision at N

This measure is used especially to evaluate RS [Lerato et al., 2015]. Given that, from the perspective of the user of a RS, it is highly desirable that at least the first recommendations be highly relevant. So, it is interesting to evaluate the proposed settings based on the precision score considering the N top ranking providers (up to the tenth) according to the usual formula for "precision at N". "Precision at N" represents the portion of the top-n documents that are relevant to the user.

$$P@N = \frac{\text{relevant items in the top } N \text{ recommended items}}{N}. \tag{2.4}$$

2.7.5 Cross-validation

Cross-validation [Refaeilzadeh et al., 2009] is a technique that is used to evaluate and compare the performance of models. This technique is used

in the case of insufficient data available for partitioning them into training and test sets. K-fold cross-validation consists in randomly partitioning the data into K equal folds. Subsequently, one of the folders is used as the test set while the remaining k-1 folds are used for training set. This operation is repeated until each unique folder is used as the test set.

2.8 Conclusion

In this chapter, we provided some background notions on the semantic Web, ontology alignment, knowledge extraction, and recommender systems. All these notions are the backbone of this thesis since we intend to propose an approach to inject domain knowledge in a recommender system for the sourcing domain.

Chapter 3

Ontology engineering for the sourcing domain

Contents

3.1 Introduction

In recent years, various companies have been moving towards the integration of ontologies within their processes to better structure their knowledge and improve the performance of their automatic processing. In this chapter, we present our domain knowledge modeling specific to the sourcing domain with the goal of reasoning on knowledge to improve the providers' recommender.

This chapter is organized as follows: we first present relevant works on ontology engineering in Section 3.2. Subsequently, we propose in Section 3.3 the main lines of our approach to build an ontology for the sourcing domain. Sections 3.4, 3.5 and 3.6 detail the three steps of our approach. Finally, Section 3.7 concludes this chapter.

3.2 Related works

Since 1996, several surveys on ontology engineering methodology have been written [Uschold et al., 1996; Fernández-López and Gómez-Pérez, 2002; Corcho et al., 2003; Iqbal et al., 2013; Stadlhofer et al., 2013; Simperl and Luczak-Rösch, 2014; Yadav et al., 2016; Kotis et al., 2020].

In this section, we describe well-known and relevant methodologies:

Uschold and King's methodology [Uschold and King, 1995]: was designed at the Artificial Intelligence Applications Institute (AIAI) of Edinburgh to build an ontology for enterprise modeling processes. It defines four steps to build an ontology: (i) identify the purpose; (ii) build the ontology; (iii) evaluate it; and (iv) document it. Three strategies for identifying the main concepts in the ontology are also presented: (i) in a top-down approach the most abstract concepts are identified first; (ii) a bottom-up approach starts by the identification of the most specific concepts, and then their generalization into more abstract concepts; and (iii) a middle-out approach starts by identifying the important concepts, and then either generalizes or specializes into other concepts.

Grüninger and Fox methodology [Grüninger and Fox, 1995]: was designed in the context of the development of knowledge-based systems using first order logic, and is based on two steps: (i) identify the main applications and scenarios for which the ontology will be used; (ii) determine the scope of the ontology using a set of competency questions.

KACTUS approach [Bernaras, 1996]: is designed in the context of the Esprit KACTUS project with the aim of studying the feasibility of knowledge reuse in complex technical systems, and the role of ontologies to support it. Ontologies are built following a bottom-up strategy. The

idea is to refine the ontology each time that the application is built. The KACTUS approach proposes three steps: (i) specification of the application; (ii) preliminary design based on relevant top-level ontological categories; (iii) ontology refinement and structuring.

METHONTOLOGY [Fernández-López et al., 1997]: created in the Artificial Intelligence Lab from the Technical University of Madrid, this methodology is used to build ontologies either from scratch, reusing other ontologies, or for re-engineering ontology [Corcho et al., 2003]. It enables the construction of ontologies at the knowledge level. This methodology divides the ontology development life cycle into: (i) specification; (ii) conceptualization; (iii) formalization; (iv) implementation; and (v) maintenance.

SENSUS [Swartout et al., 1996]: was developed by the Information Sciences Institute (ISI) natural language group to provide a conceptual structure for developing machine translators. The process of this methodology starts by extracting information from various electronic sources of knowledge. The basic ontology is manually aligned first with PENMAN Upper Model [Bateman, 1995] and ONTOS [Nirenburg and Defrise, 1992] and then with WordNET.[1] SENSUS is composed by the following steps: (i) seed identification using series of terms; (ii) manually link these seed terms to SENSUS; (iii) include all concepts from the new terms to the root of SENSUS; (iv) add all new terms that could be relevant within the domain; (v) add the entire subtree under the nodes that have a large number of paths through them; (vi) add new domain terms.

NeOn [Suárez-Figueroa, 2010; Gómez-Pérez, 2009]: is a scenario-based methodology that supports the collaborative aspects of ontology construction. It emphasizes the development of ontology networks as well as the reuse of existing ontological and non-ontological resources to the development of an ontology [Kotis et al., 2020]. This methodology presents a set of nine scenarios: (i) from specification to implementation; (ii) reusing and re-engineering non-ontological resources; (iii) reusing ontological resources; (iv) reusing and re-engineering ontological resources; (v) reusing and merging ontological resources: ontology matching tools enable ontology aligning or merging; (vi) reusing, merging and re-engineering ontological resources; (vii) reusing ontology design patterns (ODPs); (viii) restructuring ontological resources; and (ix) localizing ontological resources to translate all the terms of the ontology into another natural language.

On-To-Knowledge [Staab et al., 2001]: is based on analyzing use cases and includes the identification of goals of the knowledge management

[1] https://wordnet.princeton.edu/

tools. This methodology is composed of four steps: (i) Kick-off to capture the ontology requirements, identification of the competency questions, and study of the potentially reusable ontologies, (ii) refinement; (ii) evaluation; and (iv) ontology maintenance.

These methodologies focus on different aspects of ontology engineering [Corcho et al., 2003; Kotis et al., 2020]:

- Some methodologies are designed to build ontologies from scratch, or promote the reuse of existing ontologies (i.e. NeOn, METHONTOLOGY). KACTUS proposes to build an ontology based on an abstraction process from an initial knowledge base, while SENSUS proposes to automatically generate the ontology's skeleton from a huge ontology.
- Degree of application dependency: (i) application dependent (i.e. KACTUS and ON-To-Knowledge methodology) since the ontology is built based on a given application; (ii) semi application-dependent for example Gruninger and Fox's method and SENSUS; (iii) application independent (i.e. other methodologies) since the ontology development process is totally independent of the uses of the ontology.
- Cooperative construction: only the NeOn methodology takes this aspect into account.
- Life cycle proposition: only METHONTOLOGY and On-To-Knowledge methodologies propose a life cycle to identify the set of phases through which the ontology moves during its life.
- Strategies for identifying concepts: (i) top-down approach (i.e. KACTUS); (ii) bottom-up approach (i.e. SENSUS); and (iii) middle-out approach (i.e. METHONTOLOGY, On-To-Knowledge)

3.3 Our ontology design approach

We adapted the NeON methodology to build the *Silex* vocabulary based on the scenario of reuse, fusion, and re-engineering of resources. We identified three main questions in our ontology engineering approach:

1. What types of knowledge do we need to represent in order to reason on their representation and to improve the provider recommendation quality?
2. Which existing ontologies can we reuse?
3. Do we need to develop a new complementary ontology?

With the collaboration of *Silex* managers, and after an analysis of textual descriptions of providers and service requests, we decided to represent four types of knowledge:

1. Skills defined as a set of required knowledge and capacity to realize a daily task in a specific field [Amourache et al., 2008].
2. Occupations[2] performed by a person in a specific field.
3. Business activities[3] refer to any activity a business engages in for the primary purpose of making profit. This is a general term that encompasses all the economic activities carried out by a company during the course of business.
4. Products defined as goods, services or ideas that can be offered for sale.

We imagined scenarios relevant for the *Silex* company and based on them we identified competency questions, i.e. the questions that our ontology must enable to answer.

3.3.0.1 Scenarios

1. The user is looking for a special product or service. The ontology must link the descriptions of the service request and providers based on this need.
2. The user is looking for a special product or service. But no provider description contains directly this need. The ontology must allow to navigate through this knowledge to answer a more generic or more specific need.
3. The user wants to have an idea about a product or a service but also about related services.

3.3.0.2 Competency questions

In order to specify correctly our ontology, and with the help of the *Silex* managers, we have defined a set of competency questions [Uschold et al., 1996; Noy and McGuinness, 2000]:

1. Which skills / occupations / business activities/products appear in the description of a provider or a service request?
2. What are the links between two skills / business activities / products?
3. Which service providers have a given occupation / skill / business activity/ product?
4. What are the skills related to a given occupation / business activity?
5. What are the products related to a given occupation / business activity?

[2]https://fr.wikipedia.org/wiki/Profession
[3]https://www.investopedia.com/terms/b/business-activities.asp

Our approach for building an ontology involves:

- Identifying current metadata repositories that represent skills, occupations, products and business activities.
- Building an internal *Silex* metadata repository to represent the company's own knowledge.
- Aligning the identified metadata repositories and ontologies with the internal ones.

This is further detailed in the following sections.

3.4 Identification and reuse of existing metadata

In the context of *Silex*, our need is based on the use of domain knowledge that can be captured in a thesaurus of concepts for the procurement domain, as we exploit different types of relationships between concepts and their labels (not intentional definitions). In the rest of this thesis, we will refer to vocabulary indistinctly as a thesaurus or an ontology.

We reviewed the state of the art of existing metadata repositories and identified the most interesting ones for our context. We defined four selection criteria: (i) the source of the metadata repository; (ii) its freshness; (iii) the supported languages; and (iv) the supported formats.

We studied three types of vocabularies: (i) the first one represents skills and occupations; (ii) the second one represents the business activities; and (iii) the third one represents products.

3.4.1 Skills and occupations vocabulary

3.4.1.1 European skills, competences, qualifications and occupations

European Skills, Competences, Qualifications and Occupations (ESCO) [4] is a multilingual classification of occupations and skills, containing 17091 concepts and available in SKOS-RDF format. The main goal of ESCO is to bridge the gap between the world of education and training and the European Union (EU) labor market by identifying and categorizing skills, qualifications and occupations in the EU. ESCO provides descriptions of 2942 occupations and 13485 skills linked to these occupations, translated into 27 languages (all official EU languages plus Icelandic, Norwegian and Arabic). The hierarchical structure of ESCO is composed of five levels. We present in figure 3.1 an example of ESCO hierarchy knowledge.

[4] https://ec.europa.eu/esco/portal/home

```
Professionals
└─ Information and communications technology professionals
   └─ Software and applications developers and analysts
      └─ Systems analysts
         └─ ICT system architect
```

Figure 3.1: ESCO example.

3.4.1.2 Operational directory of trades and jobs

Operational Directory of Trades and Jobs or *Répertoire Opérationnel des Métiers et des Emplois* (ROME)[5] is a directory created in 1989 by the National Employment Agency (ANPE), today *Pôle emploi*, in France. The last version of ROME contains more than 10,000 different names of job titles grouped by families; for example, administration of information systems (*administration de systèmes d'information*), database administrator (*administrateur / administratrice de bases de données*) or production and operation of information systems (*production et exploitation de systèmes d'information*). The ROME code, consisting of a letter and four digits, is structured into four levels:

- The letter (from A to N) represents a family of occupations;
- The letter and the first two digits identify the professional field;
- The letter and the first three digits further specifies the professional field;
- The letter and the four digits, representing the ROME code, refer to the occupations as concepts.

We present in figure 3.2 an example of the ROME hierarchy knowledge. ROME is available in Excel format, but we were planning on building a vocabulary in SKOS-RDF format, therefore, as a preprocessing step, we had to transform rome into SKOS-RDF format. Our vocabualry containing 12255 concepts.

3.4.1.3 The nomenclature of Cigref's IS professions

The mission of the Association of the French large companies and public administrations or *Association des grandes entreprises et administrations publiques françaises* (CIGREF) is to develop the capacity of large companies to integrate and master digital technology. It maintains a nomenclature of

[5] http://www.pole-emploi.org/accueil/mot-cle.html?tagId=94b2eaf6-d7bd-4244-bddc-01415605563b

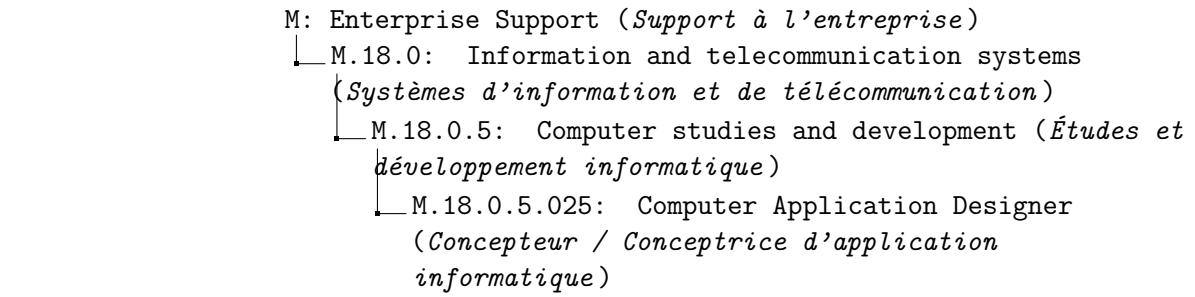


Figure 3.2: ROME example.

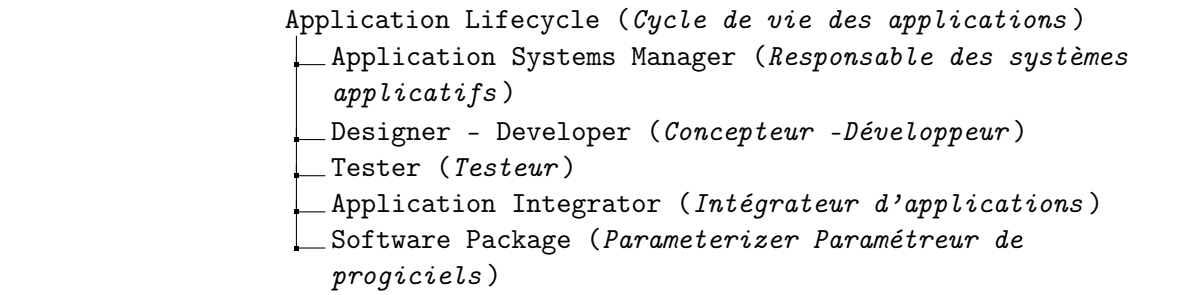


Figure 3.3: CIGREF example.

professions [6] that provides a description of existing professions in the IT departments of the large companies that are members of CIGREF. This nomenclature contains seven sub domains of computing and 36 occupation names and descriptions for each business. We built a vocabulary in SKOS-RDF format for this repository in the same way as for ROME, which contains 43 concepts. We present in figure 3.3 an example of the CIGREF hierarchy knowledge.

3.4.2 Business activity vocabulary

3.4.2.1 French activity nomenclature (NAF)

NAF [7] is a nomenclature of productive economic activities, mainly developed to facilitate the organization of economic and social information. In order to facilitate international comparisons, it has the same structure as the European nomenclature of activities (NACE), itself derived from the international nomenclature (CITI). NAF was established in 1993. The final version is NAF rev.2 was established in 2008. We chose to use NAF instead of using NACE or CITI because there is a version of NAF in SKOS-RDF format containing

[6] https://www.cigref.fr/publication-mise-a-jour-2018-de-la-nomenclature-des-metiers-si-du-cigref

[7] https://www.insee.fr/fr/information/2406147

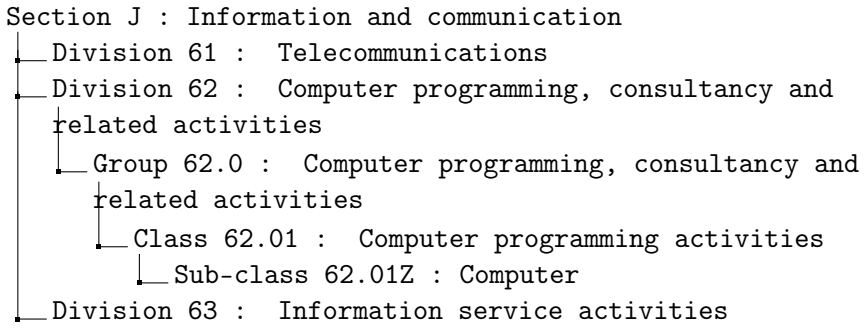


Figure 3.4: NAF example.

1735 concepts with both French and English labels. NAF rev.2 has a tree structure with five levels:

- 21 sections (1 letter), common to NAF, NACE and ISIC.
- 88 divisions (2 digits), common to NAF, NACE and ISIC.
- 272 groups (3 figures), common to both NAF and NACE.
- 615 classes (4 digits), common to both NAF and NACE.
- 732 sub-classes (4 digits and 1 letter).

We present in figure 3.4 an example of the NAF hierarchy knowledge.

3.4.2.2 United Nations standard product and services code (UNSPSC)

UNSPSC [8] is an open and international classification of business activities, goods and services, owned by the United Nations Development Programme (UNDP). The UNSPSC is an eight-digit coding system with a four-level hierarchical structure:

- Segment is the logical aggregation of families for analytical purposes.
- Family is a commonly recognized group of interrelated commodity categories.
- Class is a group of commodities sharing common characteristics.
- Commodity is a group of substitutable products or services.

[8] https://www.unspsc.org/

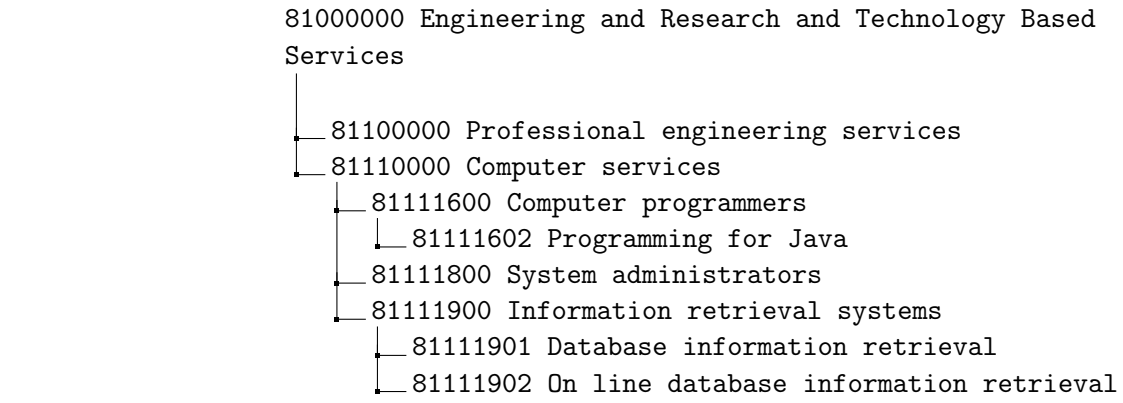


Figure 3.5: UNSPSC example.

This classification is available in 11 languages. A commercial version is available in Microsoft Excel format. In the same way as we did for ROME, we transformed the Excel file of UNSPC into a SKOS-RDF vocabulary. For that, we started by analyzing the format of this file, which contains 18 columns. For each level of the UNSPC tree, there are four columns. For example, the columns associated to the segment level are: (i) "segment" which represents the key of segment; (ii) "segment title" which represents the title of the segment; (iii) "segment definition" which presents a description of the segment; and (iv) "segment synonyms" which provides all title synonyms in all languages. We used an automatic language detector to extract the French language appellation from the synonyms columns, and we built our vocabulary in SKOS-RDF format. Our UNSPSC version contains 87470 concepts.

We present in figure 3.5 an example of the UNSPSC hierarchy knowledge.

3.4.2.3 Kompass

Kompass [9] is the author of the most extensive international classification of activities. This classification, whose original version was created in 1947, makes it possible to systematically classify companies in the 66 countries of the Kompass network, according to the products and services they provide. Its 58,000 entries make up a unique directory, continuously updated, translated into 26 languages, offering very important development perspectives. In 2014, the WF13 new version of Kompass International was built: this classification proposes a new hierarchical structure that takes into account the latest developments in the various sectors of the world economy. WF13 offers a ranking of 55,000 products and services presented in a tree of 4 levels:

[9] http://www.kompass-international.com/Corporate/home/kompass-know-how/processing-the-data/classification.html

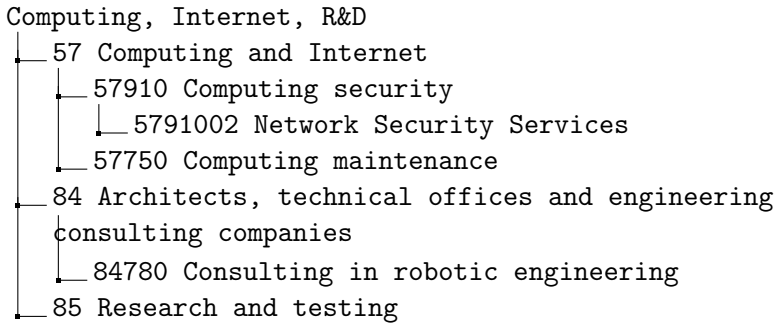


Figure 3.6: KOMPASS example.

- 15 families,
- 67 sectors (2 digits),
- 3014 branches (5 digits),
- 55450 products and services (7 digits).

The disadvantage of this classification is that it is exposed via a website only. We extracted this classification using a crawler to transform it into the SKOS-RDF format. Focusing on the IT field, we identified the family "IT, Internet and R&D" which contains 3 sectors: (i) IT and the Internet, (ii) Architects, technical offices and engineering consulting firms, and (iii) Research and testing. The Kompass vocabulary contains 1370 concepts.

We present in figure 3.6 an example of the KOMPASS hierarchy knowledge.

3.4.3 Products vocabulary

3.4.3.1 French classification of products (CPF)

The nomenclature of activities and products[10] have been developed mainly to facilitate the organization of economic and social information. It aims to classify goods and services resulting from economic activities. Each NAF code is associated with a link to the CPF to view the codes and titles of the products associated with each activity and to access the entire CPF. We transformed the Excel file of CPF into a SKOS-RDF vocabulary. This vocabulary contains 5522 concepts.

We present in figure 3.7 an example of the CPF hierarchy knowledge.

[10]https://www.insee.fr/fr/information/2493496

```
62.01.1 Computer design and development services
├─ 62.01.11 Computer design and development services for
│  applications
└─ 62.01.12 Computer design and development services for
   networks and systems
```

Figure 3.7: CPF example.

3.5 Construction of internal vocabularies at *Silex*

By comparing *Silex*'s internal repositories with the above mentioned resources, we identified new concepts specific to *Silex* that were used for sourcing purposes. In order to capture the richness of these internal repositories, we decided to build two internal vocabularies.

3.5.1 Construction of a *Silex* internal skills/ occupations vocabulary

The *Silex* platform contains a skills repository stored into its database and enriched by users. We used this text file containing 8470 terms to build a vocabulary in SKOS-RDF format. We faced two difficulties: (i) the repository mixes terms from different semantic fields such as skills, occupations, business activities, cities, countries, languages (Java, Marketing, Security agent, France, English, Paris); and (ii) the repository contains compound terms, in French and English, with spelling errors and abbreviations. So we started with a normalization step to remove all duplicates and group synonyms. We obtained 6479 terms.

Then, we used the hierarchical clustering [Berkhin, 2006; Rafsanjani et al., 2012; Dabhi and Patel, 2016] method using word embedding [Mikolov et al., 2013a] and the cosine similarity metric [Singhal et al., 2001] to identify groups of terms which are relatively homogeneous. We obtained 101 categories.

When analyzing the results of this clustering, we could clearly associate a category name to a cluster of terms such as:

- "**Test sub category**" groups the following terms: "**Tester**", "**Test and validation engineer**" ,"**Unit test**", "**User test**", "**Functional test**", "**Recruitment test**".
- "**Development sub category**" contains for example "**C++**", "**Java**", "**cackephp**", "**MangoDB**", "**Joomla**", "**SQL**", "**full stack developer**".
- "**Occupation sub-category**" contains for example "**IT pro ject manager**".

```
Computing
├── Cloud computing and security
├── consulting and computing expertise
└── Web and mobile application development
```

Figure 3.8: Silex business activities vocabulary.

- "**General services**" groups the following terms: "**gardening**", "**paint and flooring**" or "**floor covering**.

It is true that we made good progress in the construction of this vocabulary, but we had to face a real lock: How to automatically assign labels to the nodes of the formed clusters? A potential solution is to compare the distribution of words in the hierarchy (i.e the label of the node, the label of the parent node) to assign labels to each cluster [Treeratpituk and Callan, 2006]. Even if we have not dealt with this topic and preferred to focus on our recommendation approach, building an internal repository should be prioritized for the next steps to benefit of the rich vocabulary of *Silex*.

3.5.2 Construction of the internal business activities vocabulary for *Silex*

The *Silex* business activity repository is stored in a database and maintained by sales representatives. This repository contains six main business activities: (i) "**Computing**", (ii) "**Marketing and commercial**", (iii) "**Human resources management**", (iv) "**General services**", (v) "**Finance and Administrative**", and (vi) "**Industrial services**".

We extracted this repository in CSV format and built a vocabulary in SKOS-RDF format containing 90 concepts.

3.6 Ontology alignment

After building separate vocabularies as described above, we worked on the construction of a single modular vocabulary integrating them all, by aligning them with each other. Capturing these alignments helps to identify the semantic links between the different concepts (skills, occupations, business activities and products).

We started by doing manual alignment work for the specific field of computing. The manual alignment work was carried out to discover correspondences between these vocabularies: (i) ESCO to Cigref, (ii) ESCO to ROME, (ii) NAF to UNSPSC, and (iv) NAF to *Silex* business activities. This work allowed us: (i) to develop a proof of concept of our ontology-based recommendation of providers approach for *Silex* related to this domain, and

(ii) to use manual alignments as a test-bed for our automatic alignment approach presented in Chapter 4.

To define the semantic links between the concepts, we used the following properties:

- *skos:broadMatch* to define a generalization relationship between two concepts;
- *skos:exactMatch* to define a high level of similarity between two concepts (same labels);
- *skos:closeMatch* to express that two concepts are quite similar (with different labels);
- *dcterms:references* [11] to express a reference to a related resource.

Figure 3.9 explains the defined alignment approach.

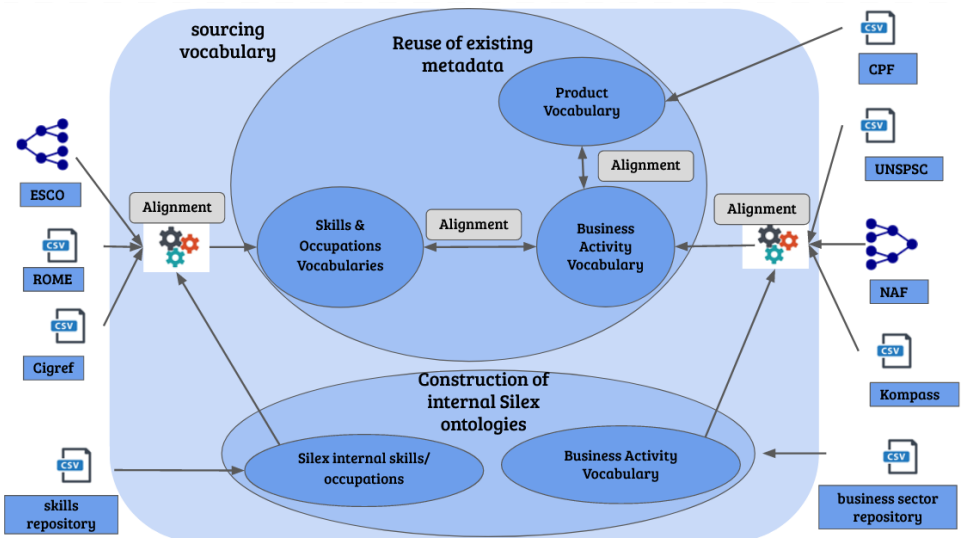


Figure 3.9: Overall knowledge engineering process to built a modular vocabulary for the sourcing domain.

3.6.1 Alignment of the skills and occupations vocabulary

To align the metadata repositories of the selected skills and occupations, we considered ESCO as the reference because of its multilingual characteristic and its completeness compared to the others. We conducted three alignment phases: (i) between ROME and ESCO (ii) between Cigref and ESCO, and

[11] https://terms.tdwg.org/wiki/dcterms:references

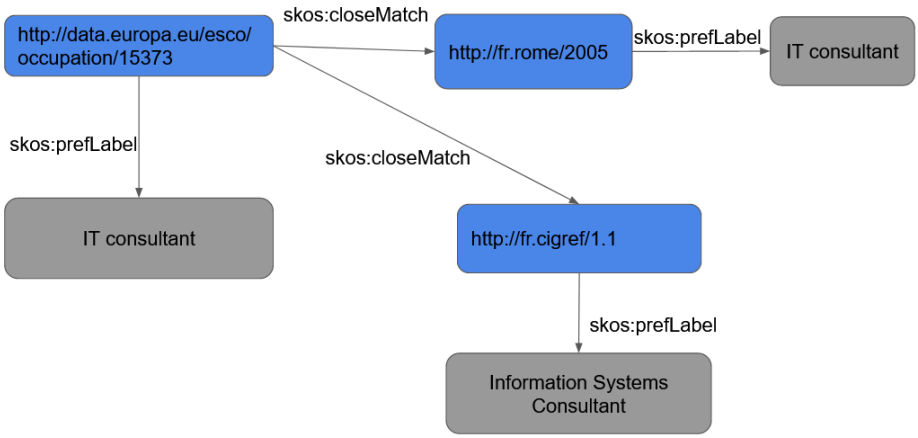


Figure 3.10: Alignment example between ESCO, ROME, and Cigref.

(iii) between *Silex_*skills and ESCO. The alignment process is mainly based on the comparison of the preferred concept labels.

We started by matching ROME/ESCO and Cigref/ESCO, and we focused on occupations data because ROME and Cigref contain only occupations.

As stated before, there are different levels of structuring for these ontologies. The highest levels can be seen as domains, while the lowest levels represent the occupations or skills. We started by aligning the occupations (lowest level) by looking for the correspondences between the concept names. To align the vocabulary domains (highest level), we defined the following rule: if there is a correspondence between the source vocabulary occupation and the target vocabulary occupation, we establish a correspondence between the target vocabulary domain and the source vocabulary domain.

For instance, as Figure 3.10 shows, we matched the ESCO concept "**IT consultant**" with the concept "**IT consultant**" of ROME and the concept "**Information Systems Consultant**" of Cigref using the property *skos:closeMatch*.

The second step of our alignment process was to match *Silex_*skills with ESCO. We can easily establish a direct match between some occupations or programming languages. For example, we match the "**IT project manager**" Silex concept with the "**ICT project manager**" ESCO concept, and we match the "**Java**" Silex concept with the ESCO concept having the same name.

Regarding the Cigref/ESCO alignment, Cigref vocabulary is limited to popular occupations from the Information Systems Departments. In spite of that, we failed to match "**Support and Assistance**" Cigref domain which contains "**Functional Assistant**" and "**User Support Technician**" occupations. Also, we could not match "**information Systems Planner**",

"**Coach agile**" or "**Product Owner**" Cigref occupations.

In the case of ROME/ESCO matching, we aligned all the occupation domains of ROME with ESCO concepts. But we failed to link all occupations of ROME. For example, for the "**Information Systems Department**" occupation domain, we could not identify a match to "**Director of the IT department**" or "**Head of IT division**" occupations. For "**Production and operation of information systems**" domain, we could not link "**IT Production Team Leader**" or "**IT Operating Assistant**" occupations.

For *Silex*_Skills/ESCO matching, we encountered several problems related to the heterogeneity of the Silex vocabulary: (i) it contains not only skills and occupations but also terms which represent sub-categories such as "**design digital**"; (ii) *Silex* details more than ESCO the different software packages and programming languages, e.g. with concepts like "**mongodb**", "**windev**", "**openerp**"; (iii) finally, the Silex vocabulary is sometimes using a commercial oriented language style, for example "**front_end developer**" or "**full stack developer**".

We obtained 229 links between ESCO and ROME and 56 links betwwen ESCO and CIGREF.

3.6.2 Alignment of the business activity vocabulary

We considered NAF as the business activity vocabulary reference, and the alignment consists of searching a correspondence between NAF concepts, Kompass concepts, and UNSPSC concepts based on their labels. We chose to proceed from the top level of hierarchies down in order to achieve the best precision.

For example, the NAF division 62 and the group 62.0 have the same name "**Programming, consulting and other IT activities**". We aligned the NAF group 62.0 with the Kompass branch 57830 "**IT audit and consulting**". We also used the *skos:narrowMatch* property to establish a generic relation between the "**Computer programming**" NAF concept and "**Language and programming software**" and "**Computer programming services**" Kompass concepts.

To align UNSPSC vocabulary to NAF vocabulary, we used not only the labels of concepts but also their definitions. We browsed all the commodities of a given class in order to define the matching between that class and NAF concepts. Therefore, the granularity of our matching is fixed at the class level. For example, we linked the UNSPSC concept "**Computer programmers**" to the NAF concept "**Computer programming services**" with the *skos:exactMatch* property. We also linked the UNSPC concept "**Software maintenance and support**" to the NAF concept "**Computer facilities management services**" with the *skos:closeMatch* property. We could not identify a matching for these classes: "**Graphic display services**", "**Art de-**

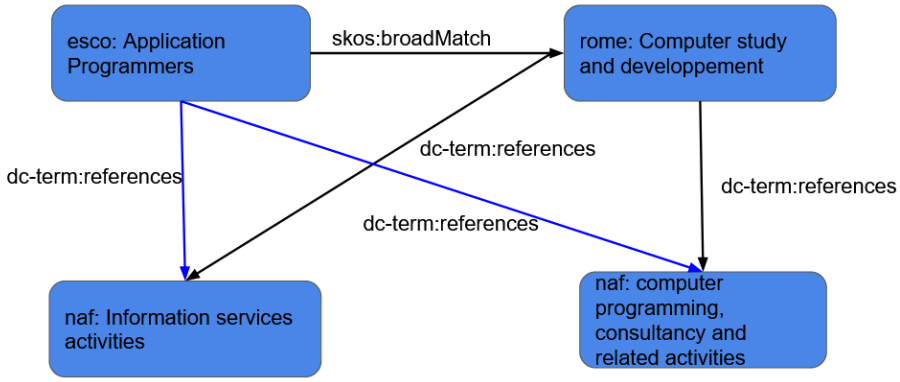


Figure 3.11: Example of ESCO-NAF alignment based on ROME.

sign services", "**Electronic mail & messaging service**", "**Data voice or multimedia network**" and "**Access management service**".

We matched *Silex* _activity to NAF. We aligned for example "**Cloud computing and security**", "**IT and IS consulting**", and "**Web and mobile application development**" concepts, but we did not match"**Design and artistic direction**", "**3 D Impression" and "Conversational Digital Assistant**" concepts.

We obtained 77 links between NAF and UNSPSC, 19 links between NAF and Silex_activity, and 11 links between NAF and KOMPASS.

3.6.3 Alignment between the skill/occupation vocabulary and the business activity vocabulary

The last step of the alignment process was the elaboration of links between the skill and occupation ontology and the business activity vocabulary. The best way to do that is to align the two reference ontologies ESCO and NAF. We used the Pôle Emploi documentation, which gives the correspondences between ROME and NAF [pôle emploi, 2017]. Having already established the links between ESCO and ROME, we thus deduced by transitivity the links between ESCO and NAF. The total number of links between these two ontologies is 34 links. For example, we had already aligned the ESCO concept "**Application Programmers**" with the ROME concept "**Computer studies and development**". The Pôle Emploi documentation establishes a correspondence between the latter and NAF divisions "**Computer programming, consultancy and related activities**" and "**Information service activities**". As a result, we have defined a correspondence between the ESCO concept and NAF divisions "**Computer programming, consultancy and related activities**" and "**Information service activities**". Figure 3.11 presents an alignment example between ESCO and NAF based on ROME.

Table 3.1: Metadata repositories.

Repository	Knowledge	Coverage domains	Format	Languages	Number
ESCO	Skill& Occupation	All domains	RDF	Multilingual	17091
ROME	Occupation	All domains	SKOS-RDF transformation	French	12255
Cigref	Occupation	Computing domain	SKOS-RDF transformation	French	45
NAF	Business sector	All domains	RDF	French English	1735
UNSPSC	Business sector	All domains	SKOS-RDF transformation	French English	87470
kompass	Business activity	Computing domain	SKOS-RDF transformation	French	1370
CPF	Product	All domains	SKOS-RDF transformation	French	5522

We obtained 34 links between ESCO and NAF.

3.7 Conclusion

In this chapter, we described our approach to design the *Silex* modular vocabulary intended to support an ontology-based approach of automatic sourcing. The resulting vocabulary makes it possible to semantically annotate text descriptions of providers and service requests based on four knowledge types: (i) skills, (ii) occupations, (iii) business activities and (iv) products, in order to automatically generate high-quality recommendations of service providers. We adopted a top-down approach, reusing metadata repositories such as ESCO, ROME, NAF, UNSCPC and CPF. Table 3.1 summarizes the existing repositories used for the sourcing domain knowledge. We adopted a bottom-up approach to build the company's internal vocabulary. The final step of aligning the various vocabularies chosen to build up the targeted modular vocabulary was first carried out manually for the computing sector only, as a proof of concept and a way to build up a test bed. Chapter 4 presents an approach that we designed to automatically align candidate vocabularies for any other sector.

Chapter 4

Ontology Alignment

Contents

4.1 Introduction

Ontology alignment plays a key role in the management of heterogeneous data sources and metadata. Many reasons can explain this: (i) there are different actors with different interests, (ii) the use of different tools, knowledge with some different levels of details [Euzenat et al., 2007], (iii) and also different methodologies followed to construct the ontologies, etc. As a result, several ontologies exist in the same or different domains with some level of heterogeneity among them. Ontology alignment is thus a crucial yet difficult task to deal with this heterogeneity and achieve interoperability on the semantic web. In this context, we address in this chapter the following research questions:

- *How can we align two ontologies?*
- *How can we define a similarity measure between the entities of ontologies?*
- *How can we refine the nature of the relationship between two entities?*

We propose a novel approach to ontology alignment based on a set of rules exploiting mainly semantic information using a similarity measure defined in the embedding space of a word embedding. The underlying assumptions behind our approach are:

- All the labels of the entities which share the same parents are close to each other in the embedding space;
- Each entity in an ontology can be represented as a cluster of its instances in the embedding space and such a cluster can be described by its centroid and its radius [Ristoski et al., 2017; Alshargi et al., 2018b,a];
- A cluster whose radius is smaller than the radius of another cluster whose centroid coincides or is very close to its centroid is likely to represent a specialization of the entity associated with the broader cluster.

Our major contributions include:

- Our capability to handle not only the equivalence relationship, but also the hierarchical relationship between entities;
- The introduction of the radius notion as a dispersion measurement of a label cluster that enables to refine the nature of the relationship (equivalence or hierarchical) between two matching entities;
- Our capability to discover rich n-m relationships between entities;

- The evaluation of our system on several open datasets from the Ontology Alignment Evaluation Initiative (OAEI)[1] benchmark and a real-world case study provided by the *Silex* company [2] and another provided by *ONISEP* [3] .

This chapter is organized as follows: Section 4.2 discusses previous works on ontology alignment. Section 4.3 describes the different steps of our ontology alignment approach. Section 4.4 reports and discusses the results of our experiments on several datasets. Section 4.5 concludes with an outline of future work.

4.2 Related work

A variety of ontology alignment techniques has been presented in the literature, and probably over a hundred different alignment systems exist to date. Due to this wide scope, we choose not to capture all research directions in this domain. Instead, we focus in this section on giving an overview of alignment techniques with some references of systems. Several surveys on ontology alignment techniques have been written [Ardjani et al., 2015; Euzenat et al., 2007; Kalfoglou and Schorlemmer, 2003; Otero-Cerdeira et al., 2015; Shvaiko and Euzenat, 2005; Rahm and Bernstein, 2001; Doan and Halevy, 2005]. Most of these surveys focus on input and process dimensions to classify the ontology alignment techniques. Doan and Halevy [2005] consider both input and process dimensions and differentiate in their classification between: (i) rule-based techniques that exploit schema-level information in specific rules; and (ii) learning-based techniques that exploit data instance information with machine-learning or statistical analysis. However, Rahm and Bernstein [2001] analyze the two dimensions in a different way. For the input dimension they distinguish between instance classification matchers (i.e. exploiting information from the TBox) and schema classification matchers (i.e. exploiting information from the ABox). For the process dimension they introduce classification axes such as element vs structure or linguistic vs constraint-based. But the most complete and extensive classification of ontology alignment techniques available to date is probably the one proposed by [Euzenat et al., 2007] depicted in Figure 4.1.

This classification can be read in a top-down way focusing on the granularity of the matcher and the interpretation that the different techniques offer to the input information, or in a bottom up way focusing on the origin and the kind of input information used by the matching techniques. In the top-down interpretation, the matching techniques can be classified in the first level as:

[1] http://oaei.ontologymatching.org/
[2] https://www.Silex-france.com/Silex/
[3] http://www.onisep.fr/

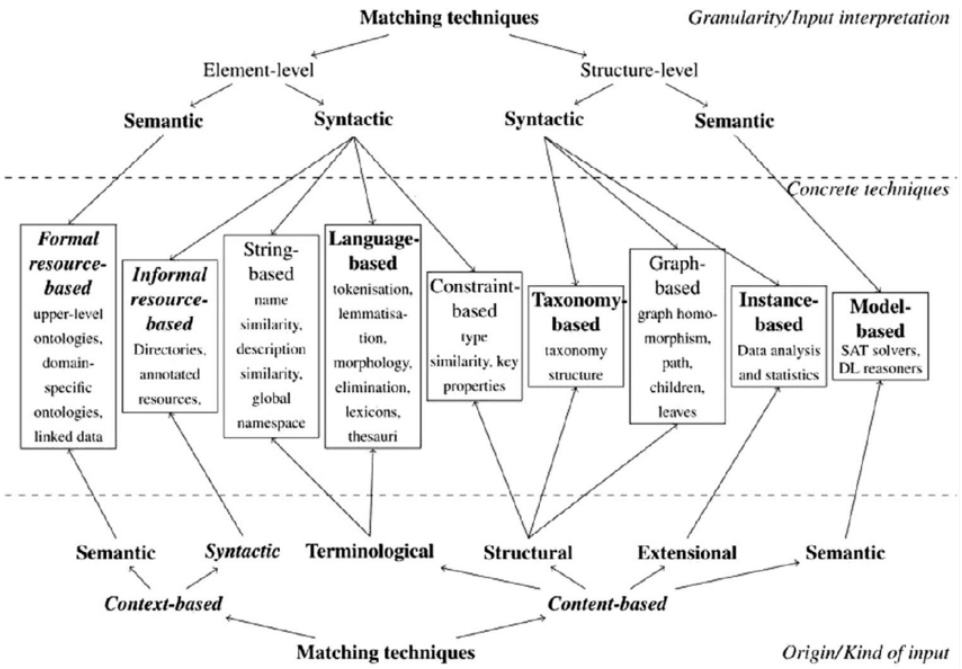


Figure 4.1: Matching techniques classification. Source [Euzenat and Shvaiko, 2013].

- Element-level techniques, which find out correspondences by considering the entities in isolation.
- Structural-level techniques, which rely on the analysis of the neighbourhood of two entities to determine their similarity.

In the second level of the top-down interpretation, the matching techniques include:

- Syntactic techniques, which limit their input interpretation to the instructions stated in their corresponding algorithms.
- Semantic techniques, which interpret their input using formal semantics.

In the bottom-up interpretation, the matching techniques can be classified in the first level as:

- Content-based techniques, which focus on the internal information of the two ontologies.
- Context-based techniques, which take into account external information that may come from relations between ontologies or external resources [Otero-Cerdeira et al., 2015].

Most researches on ontology alignment fall into the content-based category, which is divided into four groups or feature approaches, ranging from lexical information to semantics information, passing through structural and extensional information [Kolyvakis et al., 2018].

4.2.0.1 Lexical information

Lexical information presents an important source of information to ontology alignment systems. This type of information is deduced on the Element-level and is based on computing the similarities between the lexical information of the entities (i.e. names, labels, descriptions, comments). There are three main categories of similarity measures to compare two strings [Cheatham and Hitzler, 2013; Stoilos et al., 2005; Nguyen et al., 2013; Gomaa et al., 2013]:

- String-based similarity measures: consider a string as a sequence of characters: (i) The Hamming distance is a simple way to compare two strings based on counting the number of positions in which two strings differ [Euzenat et al., 2007]. (ii) The edit distance (also called the Levenshtein distance) is a basic character-based metrics that represents the minimal cost of editing operations (i.e insertion, deletion, and substitution) to be applied to one string in order to obtain the other one. This metric is very popular in ontology alignment systems such as RIMOM [Li et al., 2008], ASMOV [Jean-Mary et al., 2009] and AgreementMaker [Cruz et al., 2009]. (iii) The character-based metrics, also called the Jaro distance, measures the similarity between two strings based on the number of common characters that are present in them. The main limitation of this technique is that it does not allow to discover the equivalent entities described by different terms (synonyms), while different concepts described by equal terms (homonyms) will mistakenly be detected as a perfect match [Granitzer et al., 2010]. COMA [Do and Rahm, 2002] and COMA++ [Aumueller et al., 2005], OLA [Euzenat and Valtchev, 2004], Anchor-Prompt [Noy and Musen, 2001], S-Match [Giunchiglia et al., 2004] are examples of systems that use string-based metrics for alignment.

- Token-based similarity measures: consider a string as a vector, which makes it possible to apply metric space distances: (i) The Term Frequency-Inverse Document Frequency (TF-IDF) [Cohen et al., 2003] measures the relevant of a word in the document; it is defined as the product of the frequency of the word in the document (TF) and its importance according to its distribution and use in the document set (IDF). (ii) The Jaccard similarity measure [Hadjieleftheriou and Srivastava, 2010] is defined as the ratio between the number of common characters between two strings and the total number of characters. (iii) The Euclidean distance represents the geometric distance between two

data points in an n-dimensional space, (iv) The Manhattan distance calculates the distance between two real-valued vectors. (v) The cosine similarity is a measure of similarity between two vectors of an inner product space that measures the cosine of the angle between them [Gomaa et al., 2013]

- Hybrid similarity measures: combine string-based and token-based approaches: (i) SoftTF-IDF [Cohen et al., 2003] determines similar token pairs by improving the TF-IDF method by a character-based metric (Jaro), (ii) TagLink [Camacho and Salhi, 2006] first calculates a similarity score for each pair of tokens by comparing the characters of one token to those of the other token, and then computes a global similarity score which is calculated for both strings by comparing the tokens in one string to those in the other string, and (iii) [Monge et al., 1996] use a character-based metric (Jaro). For each token of the first string, they look for the closest token in the second string and the corresponding score. The global similarity score between the two strings corresponds to the average value of these scores.

4.2.0.2 Structural information

The lexical information is not enough to discover the matching when the vocabulary of ontologies differs. Hence, many ontology alignment systems consider the structure level for this. Ontology is represented as a kind of graph. The position of the entities in the graph and their relations with others entities can be a very good source of information to be analyzed and used to discover the similarity structure. Among the systems which use this kind of information, we can mention: Yam++ [Ngo and Bellahsene, 2012], MEDLEY [Hassen, 2012], Cupid [Madhavan et al., 2001], AnchorPrompt [Noy and Musen, 2001], COMA [Do and Rahm, 2002], OLA [Euzenat and Valtchev, 2004], QOM [Ehrig and Staab, 2004], RiMOM [Li et al., 2008], and there are many others.

4.2.0.3 External information

Despite the fact that lexical and structural information are widely used in ontology alignment, these techniques suffer from their weakness in capturing the semantics of lexical information of entities. To overcome this problem, many systems consider linguistic-based similarities: it is the case of AROMA [David, 2007], Falcon [Jian et al., 2005], OLA [Euzenat and Valtchev, 2004], Cupid [Madhavan et al., 2001], COMA [Do and Rahm, 2002]. This technique involves exploiting an auxiliary resource, such as WordNet, to add lexical relationships (e.g. synonym, antonyms, hypernyms or hyponyms) to the system [Mohammadi et al., 2018]. This information not only improves the alignment quality, but also allows to define the type of relationship, such

as equivalence or generalization [Granitzer et al., 2010]. There are three methods to calculate similarities using WordNet [Lin and Sandkuhl, 2008]: (i) an edge-based method, which estimates the semantic relatedness between two concepts in WordNet by accumulating the lengths of all edges on the shortest path to quantify the semantic similarity, (ii) an information-based method, which estimates the semantic similarity between two concepts using information contained in related nodes (i.e., related concepts) in WordNet, and (iii) a hybrid method, which combines the information-based method and the structural information (i.e edge-based method) from WordNet to estimate the semantic similarity between words. The main drawbacks of this approach are: (i) that thesauri for languages other than English are generally of poor quality or simply not available; (ii) specialized application areas require a domain-specific thesaurus.

4.2.0.4 Semantic information

If it is true that the use of an auxiliary resource can resolve part of the synonymy problem when searching for similarities between entities, auxiliary resources still suffer from the incompleteness and non-exhaustiveness of their entries. For that, word embedding techniques are now used more and more in the ontology alignment task [Zhang et al., 2014; Vieira and Revoredo, 2017; Kolyvakis et al., 2018; Lastra-Díaz et al., 2019]. The main drawback of semantic word embedding is that it tends to coalesce the notions of semantic similarity and conceptual association [Hill et al., 2015], especially because they depends on the corpus from which this embedding is derived. Still, word embedding has the potential to bring significant value to ontology matching given the fact that a great deal of ontological information comes in textual form [Kolyvakis et al., 2018].

The first approach that explored word embedding in the ontology alignment task is described by [Zhang et al., 2014]. The authors proposed a hybrid method to combine word embedding and the edit distance together. The matching strategy is to consider the maximum similarity, i.e to return for every entity in the source ontology the most similar entity in the target ontology. Nkisi-Orji et al. [2018] introduce a classifier-based approach for ontology alignment which combines string-based similarity, semantic similarity, and semantic context. Word embedding was used to generate semantic features for a random forest classifier. Kolyvakis et al. [2018] use information from ontologies and additional knowledge sources to extract synonymy and antonymy relations. These information are then used to refine and adapt pre-trained word vectors to compute the similarity distance between entities.

Schmidt et al. [2018] compare two similarity measures for synset disambiguation: (i) the Lesk measure [Lesk, 1986] and (ii) the distance between word embedding to match domain and top-level ontologies. Based on their experiments, the authors show that the results obtained using word embed-

ding are better than the results obtained with Word Sense Disambiguation. Gromann and Declerck [2018] use a multilingual word embedding for multilingual ontology alignment. Alshargi et al. [2018b,a] extend the state of the art by providing a framework containing three distinct tasks related to the individual aspects of ontological concepts: (i) the categorization aspect, (ii) the hierarchical aspect, and (iii) the relational aspect. Several intrinsic metrics are proposed for evaluating the quality of the embedding. Furthermore, multiple experimental studies were run to compare the quality of the available embedding models. This work highlights that (i) there is no single embedding model which shows superior performance for all tasks, and (ii) that the embedding learned from the knowledge graph (i.e. DBpedia) does not have a higher quality in comparison to the embedding learned from unstructured data (i.e. Wikipedia).

The semantic information category also contains other techniques than word embedding for the alignment task, by reducing the graph matching problem to pairwise node matching problems solved through the validation of a logical formula using a SAT solver. Among the systems which use this approach, let us cite CtxMatch [Bouquet et al., 2006] and S-Match [Giunchiglia et al., 2004].

When compared to the state of the art, we propose a hybrid approach combining three types of information: (i) lexical, (ii) structural, and (iii) semantic information to align ontologies. Our first challenge was to fit with the real-world use cases of the *Silex* company. The analysis of the *Silex* data showed that the labels of the entities of ontologies to be aligned are not very close at the lexical level. Therefore, string-based metrics are not very useful in this case. Then we moved towards word embedding. We experimented training our own embedding model, but we got poor results as the available corpus is not rich enough. Finally we decided to use the fastText model as it is the only model that provides word embedding for French. Based on the results obtained by Schmidt et al. [2018], which prove that word embedding preformed this task better than Word Sense Disambiguation, we decided to ignore the use of Word Sense Disambiguation in our approach.

Additionally, we considered extracting the semantics of the concepts based on the structure of the ontology. According to Aristotle's fundamental predictive theory, the semantics of a concept is mainly defined by the difference between this concept and its genus, or more generally its ascendants in the ontology [Parrochia and Neuville, 2014]. Therefore, in the ontology alignment literature, several works use information associated to more general concepts when searching matchings between two concepts, as this generalization of concepts is bringing more context. In our approach, we also consider taking into account the specialization of concepts when computing matchings, considering that more specific concepts will also bring additional context and semantics. To the best of our knowledge, there is no previous work considering this information in the ontology alignment process.

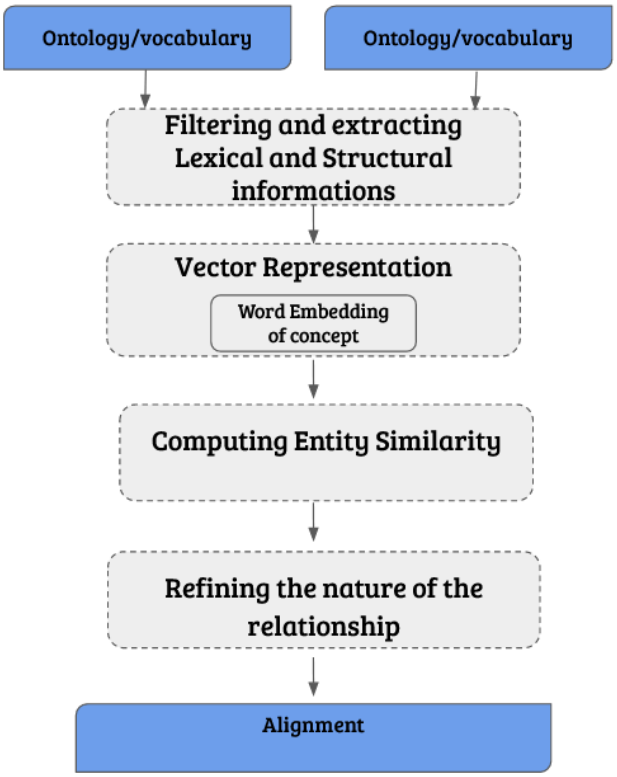


Figure 4.2: Workflow of the proposed ontology alignment approach.

4.3 Overview of our ontology alignment approach

4.3.1 Problem statement

The goal of ontology alignment is to discover the relationships between entities of ontologies.

Our alignment process, illustrated in Figure 4.2, is a hybrid approach combing lexical information, structural information and semantic information expressed in the embedding space to refine the nature of the relationship between entities. In the rest of this section, we detail the four successive steps of our approach.

We consider indifferently RDFS, OWL or SKOS vocabularies, and two languages, namely French and English. The language must be chosen at the beginning of the alignment process to ensure that the right word embedding model is selected.

4.3.2 Extracting lexical and structural information from ontologies

The first step of our approach is to extract lexical information and structural information from the ontologies to be aligned. To achieve this, the two ontologies are parsed with rdflib and queried with the SPARQL query shown in Listing 4.1.

Lexical information is extracted from the values of the properties `rdfs:label` for RDFS or OWL ontologies or `skos:prefLabel` for SKOS vocabularies.

Structural information is captured by associating the labels of all child entities to their parent entities, considering `rdfs:subClassOf` or `rdfs:subPropertyOf` properties instead of `skos:broader`. As a result, we consider clusters of entities specializing the root entity in each cluster.

Listing 4.1: SPARQL query to extract lexical and structural information from a SKOS vocabulary

```
SELECT ?uri ?label
       (group_concat
       (DISTINCT ?mid_label; separator=":")
        AS ?lineage)
WHERE {
        ?uri skos:prefLabel ?label
         FILTER (lang(?label)='fr')
        ?uri ^skos:broader* ?mid.
        ?mid skos:prefLabel ?mid_label.
        FILTER (lang(?mid_label)='fr')
      } GROUP BY ?mid ORDER BY count(?label)
```

Let us illustrate it using the hierarchy of Figure 4.3 as an example:

- lexical_information(#61) = {Telecommunications}
- structural_information(#61) = {Telecommunications, Wired telecommunications activities, Wireless telecommunications activities, Satellite telecommunication activities, Other telecommunications activities}.
- lexical_information(#J) = {Information and communication}.
- structural_information(#J) = {Information and communication, Publishing activities, Computer programming, consultancy and related activities, Telecommunications, Wired telecommunications activities, Wireless telecommunications activities, Satellite telecommunication activities, Other telecommunications activities}.

4.3.3 Computing word embedding representations

Based on the extracted information, we compute the word embedding representation of entities. We define two types of vector representations: (i) the

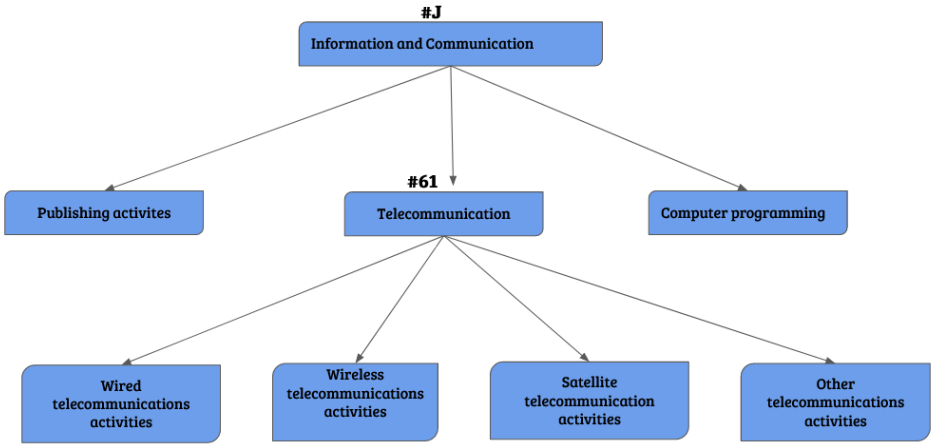


Figure 4.3: An example of a hierarchy of concepts.

vector representation of an entity (lexical information) and (ii) the vector representation of a cluster of entities (structural information).

We use the pre-trained word vectors for French and English, learned using fastText[4] on a Wikipedia dump. The French model contains 1,152,449 tokens, and the English model contains one million tokens. Both are mapped to 300-dimensional vectors [Mikolov et al., 2013b].

A pre-processing step is required to convert words to lower case and remove all stop words.

The process of computing the vector representation of the entities is similar to creating the vector representation of sentences since in several cases the label of an entity is composed of multiple words. So the vector representation of the entity is computed by averaging the word embedding vectors along each dimension of all the words contained in its label and occurring in the dictionary:

$$entityWordEmbedding(c) = \frac{1}{n}\sum_{i=1}^{n} w_i, \tag{4.1}$$

where n is the number of words in the dictionary occurring in the label of an entity c and $w_i \in \mathbb{R}^{300}$ denotes the word embedding vector of the ith such word (if a word in a label does not appear in the dictionary, it is just ignored).

The vector representation of a cluster of entities is constructed by averaging the word embedding vector representations of the entities belonging to it:

[4]https://fasttext.cc/docs/en/pretrained-vectors.html

Listing 4.2: Pseudo-code to search for matching entities

```
input: source ontology O1,
       target ontology O2,
       threshold_sim
output: list of correspondences
  list=null
  for each e1 in O1 do
    for each e2 in O2 do
        sim=cosine_sim{O1,O1}
        if sim> threshold_sim then
            list.append (e1,e2, sim)
        end if
    end for
  end for
```

$$clusterWordEmbedding(cl) = \frac{1}{k}\sum_{i=1}^{k} entityWordEmbedding(c_i), \quad (4.2)$$

where c_i is an entity in the cluster cl and $k = |cl|$.

4.3.4 Searching for matching entities

The semantic similarity between an entity of the source ontology and an entity of the target ontology is calculated by considering their vector representations. The common similarity metric for embedding is the cosine similarity measure. We consider that a correspondence exists between two entities when the cosine similarity between them is bigger than a given threshold. Our algorithm aims at collecting all the possible correspondences between entities to propose many-to-many mappings (i.e. one entity from one ontology can correspond to more than one entity in the other ontology). Listing 4.2 shows the pseudo-code of our algorithm to discover the correspondences.

4.3.5 Refining the nature of the relationship between two matching entities

At this stage, for each entity in the source ontology we have a list of matching entities in the target ontology. We must now decide of the nature of the relationships holding between entities of the source and target ontologies: an equivalence relationship or a hierarchical relationship depending on the degree of similarity between two matching entities, considering the clusters of which they are the root.

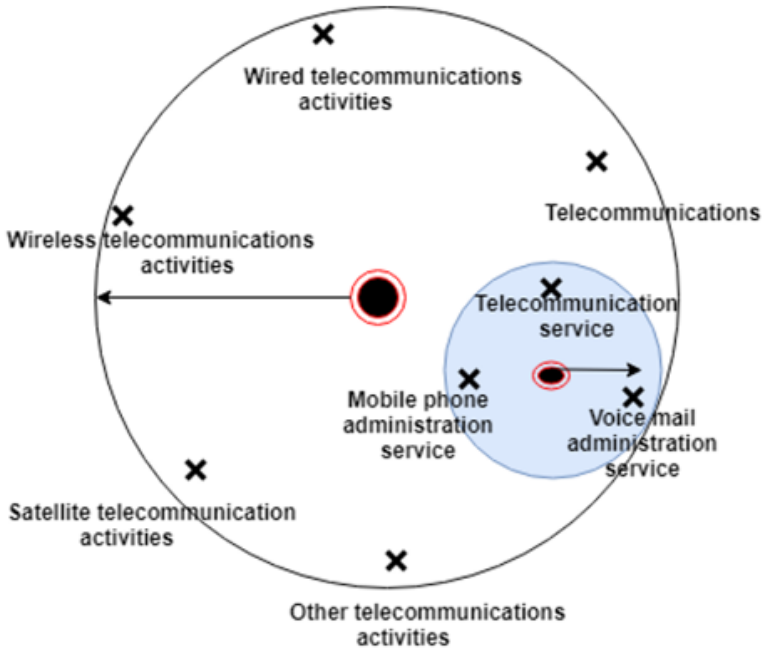


Figure 4.4: Two example clusters of entities, one included into the other.

More precisely, the relationship between two matching entities e_1 and e_2 is refined by comparing the radii of their respective embedding vector clusters, computed by taking into account the hierarchical structure of the two ontologies: The radius of a cluster is the maximum distance between the centroid of the cluster and all the other entities in the cluster. We define the radius of a cluster of entities as the standard deviation of their cosine dissimilarity with respect to the centroid:

$$radius = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(1 - \frac{w_i \cdot \overline{w}}{|w_i| \cdot |\overline{w}|}\right)^2}, \tag{4.3}$$

where $w_i \in \mathbb{R}^{300}$ is the vector representation of the i^{th} entity in the cluster, N is the size of the cluster, and $\overline{w} \in \mathbb{R}^{300}$ is the centroid of the cluster, defined as

$$\overline{w} = \frac{1}{N}\sum_{i=1}^{N} w_i.$$

Figure 4.4 shows two example clusters associated to entity *Telecommunication* (the bigger circle) and the entity *Telecommunication service* (the smaller circle). To define the type of the relationship we compare the radii of

two matching clusters. These two clusters are formed mainly using structural information. We suppose that the cluster whose result has the smallest average distance between a label and the centroid is in broader relation with the cluster which has the largest radius. As shown in Figure 4.4, the blue circle (which represents the cluster of telecommunication service, voice mail administration service, and mobile phone administration services) is in broader relation with the big circle (which represent the cluster including telecommunications, wired telecommunications activities, and satellite telecommunications activities).

We define the two following rules to identify the relationship holding between two similar entities:

$$|radius(e_1) - radius(e_2)| < 0.1 \Rightarrow e_1\ closeMatch\ e_2 \tag{4.4}$$

$$\begin{aligned} radius(e_1) - radius(e_2) > 0.1 \Rightarrow e_1\ narrowMatch\ e_2 \\ \wedge e_2\ broadMatch\ e_1 \end{aligned} \tag{4.5}$$

In particular, the first condition above is trivially satisfied when both e_1 and e_2 are leaf nodes of their respective ontologies and their radii are both zero.

We represent equivalence relationships by using *owl:sameAs* properties when aligning RDFS or OWL vocabularies and *skos:closeMatch* properties when aligning SKOS vocabularies. We represent hierarchical relationships by using *rdfs:subClassOf* or *rdfs:subPropertyOf* properties for RDFS or OWL vocabularies and *skos:broader* and *skos:narrower* properties for SKOS vocabularies.

4.4 Experiments

In this section we describe the experiments conducted to evaluate the above described proposed approach to ontology alignment: the datasets we considered, the experimental protocol we adopted and the results of these experiments.

4.4.1 Datasets

We experimented our proposed approach to ontology alignment on ontologies coming from a public benchmark and two specific use cases described in the following.

4.4.1.1 Experiments on task-oriented complex Alignment on conference Organization

We experimented our approach on the conference complex alignment benchmark [Thieblin, 2019] for ontology merging. This benchmark has been

Table 4.1: Number of entities by type for each ontology.

Ontology	Classes	Object properties	Data properties
cmt	30	49	10
conference	60	46	18
confOf	39	13	23
edas	104	30	20
ekaw	74	33	0

constructed within the framework of the OAEI and it contains 57 correspondences and five ontologies (*cmt*, *conference*, *confOf*, *edas*, *ekaw*) available in OWL format. Table 4.1 summarizes the number of entities by type contained in these ontologies [Thiéblin et al., 2018].

4.4.1.2 Silex use case

As described in Chapter 3, an ontology engineering work was carried out in the Silex company, including a manual ontology alignment task to establish correspondences between targeted referentials in the computing sector:
(i) ESCO to Cigref, (ii) ESCO to ROME, (ii) NAF to UNSPSC, (iv) NAF to Silex_activity.

Table 4.2 presents the number of concepts in each of the modules building up the `Silex` vocabulary for the computing sector, and Table 4.3 presents the number alignment per relation. We consider the set of the manually stated alignments as a test-bed for the automatic alignment approach we propose.

Table 4.2: Number of concepts for the Silex ontology for the computing domain.

Skills and Occupations		Business activity	
Ontology	Number	Ontology	Number
ESCO	160	NAF	53
ROME	117	UNSPSC	153
Cigref	42	Silex	14

4.4.1.3 ONISEP use case

ONISEP (Office national d'information sur les enseignements et les professions) is a State operator that reports to the Ministry of National Education and Youth and the Ministry of Higher Education, Research and Innovation. As a public publisher, ONISEP produces and distributes all information on training and trades. It also offers services to students, parents and educational teams. In this context, ONISEP provided us with an occupation directory

Table 4.3: Number of relation types between concepts for the Silex ontology for the computing sector.

Ontologies	Relation types	Number
ESCO to ROME	Close	68 links
	Hierarchical	33 links
ESCO to Cigref	Close	24 links
	Hierarchical	31 links
NAF to UNSPSC	Close	21 links
	Hierarchical	54 links
NAF to Silex	Close	3 links
	Hierarchical	7 links

in XML format, and the goal was to align it with ROME. The ONISEP vocabulary contains 5325 concepts and the ROME vocabulary contains 12255 concepts. We started by transforming the ONISEP vocabulary into a SKOS vocabulary then we applied our approach to align ONISEP and ROME. A gold standard, composed of 290 links and produced by an expert, is used for the evaluation of our automatic alignment approach. It contains 259 close relations and 31 hierarchical relations.

4.4.2 Evaluation protocol

The performances of our approach are measured by calculating precision, recall and F-measure [Ochieng and Kyanda, 2018]. In addition to this state-of-the-art evaluation method and taking into account the fact that our system was not designed to achieve a fully automatic sourcing process but rather to support end-users responsible for the sourcing task, by presenting a list of possible matches, we defined another evaluation method assuming that if a system is able to propose a list of k best possible matches which includes the correct match, we consider that the matching is correct. This way of evaluation does not only concern the precision metric but also the recall and F1 metrics since the correspondence is no longer considered as False Positive but as True Positive. We conducted the parameter learning (i.e threshold) through 5-fold cross validation.

4.4.3 Results and discussion

4.4.3.1 Experiments on task-Oriented complex alignment on conference organisation

We compared our matching results with the results of three state-of-the-art complex ontology matchers that were evaluated in [Thiéblin et al., 2018], namely:

Table 4.4: Evaluation of our approach on the OAEI benchmark.

Systems	Precision	Recall	f-measure
Our system with standard evaluation methods	0.32	**0.31**	**0.27**
Our system with custom evaluation methods	0.70	**0.43**	**0.51**
Ritze *et al.* 2009	0.30	0.13	0.19
Ritze *et al.* 2010	0.83	0.09	0.18
Jiang *et al.* 2016	0.09	0.11	0.10

Table 4.5: Evaluation of our approach on real world data from the Silex and ONISEP use cases.

Dataset	Evaluation	Threshold	Precision	Recall	F1
ESCO-ROME	Standard	0.85	0.49	0.74	0.58
	Custom	0.7	0.99	0.94	0.96
ESCO-Cigref	Standard	0.8	0.51	0.72	0.59
	Custom	0.8	0.92	0.72	0.80
NAF-UNSPSC	Standard	0.8	0.40	0.71	0.50
	Custom	0.7	1	0.95	0.97
ONISEP-ROME	Standard	0.87	0.42	0.73	0.52
	Custom	0.7	1	0.88	0.93

1. the system presented in [Ritze et al., 2009]and implementing a rule-based approach mostly relying on string similarity;

2. the system presented in [Ritze et al., 2010] and implementing another rule-based approach using linguistic evidence; and

3. the KAOM system presented in [Jiang et al., 2016b] and using a probabilistic framework based on Markov Logic networks.

We searched the literature for other, more recent ontology alignment systems evaluated against the same benchmark, but we could not find any, probably due to the novelty of the benchmark. Table 4.4 shows that our system clearly outperforms the others on this benchmark, with an F1 of 0.27 and we can reach an F1 of 0.51 using our evaluation method, confirming the interest of looking at clusters of entities in an embedding space both to establish correspondences between them and to resolve the nature of their relations.

4.4.3.2 Silex and ONISEP use cases

Table 4.5 presents the results of our system in the real world data from *Silex* and ONISEP use cases. For the *Silex* data, the F1 value is around 0.5 and

we can reach an F1 ranges between 0.8 and 0.97 using our evaluation method. For the ONISEP data, the F1 value is 0.52 and we can reach an F1 of 0.93 using our evaluation methods.

We conducted some additional experiments in which we add the parent label to the label of a concept. We decided to conduct these experiments because we noted that several state-of-the-art proposals [Gracia and Mena, 2012] have been made on the basis of such a bottom-up approach instead of a top-down approach. The experiment shows that the use of this information severely decreases the performance of our alignment system. For example, the F1 value when matching NAF and UNSPSC decreases from 0.50 to 0.11, and when matching ESCO and cigref it decreases from 0.60 to 0.1.

Although it looks like a dramatic step ahead with respect to the state of the art, our system still has much room for further improvement. There are four main issues that could be addressed:

1. The cosine similarity between some entities that should be matched is much lower than the matching threshold and as a consequence these matches are ignored. For example the cosine similarity between 'chairman' and 'demo chair' is 0.37.

2. Our system is not designed to test hierarchical relations between two leaf nodes. This type of relationship must pass through the structural information to calculate the radius and, thus, infer the relationship. For example, in the benchmark, 'country' and 'location' are two leaf nodes that have been matched by `rdfs:subClassOf`.

3. Based on Equations 4.4, our system can assign an equivalence relation instead of a hierarchical relation because the threshold of the difference of radius between two classes is smaller than 0.1.

4. The quality of the embedding space depends on the context of the data and the similarity between the training data and the ontology data. Therefore, the quality of our system is tightly dependent on the embedding model.

4.5 Conclusion

In this chapter we presented a novel approach of ontology alignment, based on measuring the clusters of labels in an embedding space to refine relations in ontology alignment. We reported the results of our experiments on multiple datasets: (i) the OAEI conference complex alignment benchmark; (ii) the real-world use case encountered by the Silex company, namely matching skills and competences from several ontologies in the computing field; and (iii) the real-world use case encountered by the ONISEP, namely matching occupations between the ONISEP and the ROME vocabularies. These experiments

show that our approach outperforms state-of-the-art approaches and is well suited to real world use cases, where the goal would be to propose possible alignments to experts that should be validated, as it is the case for *Silex* or ONISEP. In the next chapter, we present our approach for semantically annotating descriptions of providers and service requests and recommend relevant providers for a given service request, based on the aligned *Silex* vocabulary.

Chapter 5

Named Entity Recognition and Linking

Contents

5.1 Introduction

In a number of areas, companies are often faced with the task of dealing with large amounts of textual data. Automating knowledge extraction can help to accelerate the processing of data and this by giving the machines the possibility to execute certain tasks. In this chapter, we report our knowledge extraction approach which is composed of: (i) a named entity recognition (NER) approach based on Bi-LSTM-CRF architecture able to analyze textual descriptions (service providers and service requests) and extract the relevant parts of the text that summarize a provider's offer/ a request need (such as services, products, occupations, skills); (ii) a named entity linking (NEL) algorithm based on semantic similarity to link the extracted entities from the descriptions of service requests and providers with the concepts in the sourcing vocabulary.

In this chapter, we address the following research questions:

- Which is the best approach to extract knowledge from short texts?
- Which types of embedding must we use to extract relevant knowledge in our case?
- How can we link the extracted entities with our sourcing vocabulary?

This chapter is organized as follows: Section 5.2 presents the related works for NER and NEL. Section 5.3 describes our knowledge extraction approach. Section 5.4 describes our data and our implementation. Section 5.4.2 reports and discusses the results of our experiments. Section 5.5 concludes with an outline of future work.

5.2 Related Work

5.2.1 Named entity recognition

In this section, we focus on the related works for named entity recognition. In the literature, there are three common techniques for the NER task [Yadav and Bethard, 2019; Li et al., 2018; Nguyen et al., 2016; Li et al., 2020]: (i) knowledge-based systems; (ii) feature-engineered supervised systems; and (iii) feature-inferring neural network systems.

5.2.1.1 Knowledge-based systems

Knowledge-based systems rely basically on lexical resources and specific domain knowledge. They are called unsupervised systems as they do not need annotated training data. These systems have the advantage of having a high precision, but they present two main drawbacks: (i) most of the time, the recall is low due to incomplete dictionaries; and (ii) domain experts are needed to construct and maintain the knowledge vocabulary.

5.2.1.2 Feature-engineered supervised systems

In these systems, the NER task can be seen as a multi-class classification or sequence labelling task. Given inputs and their expected outputs, supervised systems learn how to make predictions.

Feature-engineered supervised systems are based on two main approaches: (i) an approach based on the representation of each training example. Each word in the text is represented using one or more features like word-level features (e.g. case, morphology and POS), list lookup features (e.g. Wikipedia gazetteer and DBpedia gazetteer), and corpus feature (e.g. local syntax and multiple occurrences); (ii) an approach based on a machine learning to learn a model to recognize similar patterns from unseen data. Among the common machine learning systems used for NER, we can cite Hidden Markov Models (HMM) [Eddy, 1996], Support Vector Machines (SVM) [Hearst et al., 1998], Conditional Random Fields (CRF) [Lafferty et al., 2001], and decision trees [Quinlan, 1987].

IdentiFinder [Bikel et al., 1998], is the first NER system based on HMM to identify names, dates, time expressions and numerical quantities. Malouf [2002] uses Maximum Entropy (ME) with multiple features like capitalization and a list of first names collected from various dictionaries. McNamee and Mayfield [2002] train SVM classifiers using 1000 language-related features and 258 orthography and punctuation features. McCallum and Li [2003] propose a NER system using a feature induction method for CRF.

5.2.1.3 Feature-inferring neural network systems

There are many existing taxonomies for feature-inferring neural network systems in the literature [Yadav and Bethard, 2019; Nguyen et al., 2016]. Here we adopt the presentation of [Li et al., 2020] because it seems to us the most structured and understandable, especially with the distinction of three steps. According to [Li et al., 2020], a NER feature-inferring neural network system or a NER Deep Learning system can be broken down into three steps: (i) distributed representations for input; (ii) context encoder; and (iii) tag decoder to predict tags.

5.2.1.3.1 Distributed representations for input

Word-level representation [Collobert and Weston, 2008] is one of the first neural network architectures for NER. This system uses feature vectors constructed from orthographic features (e.g., capitalization of the first character), dictionaries and lexicons. With the advent of word embedding, Collobert et al. [2011] propose a semi-supervised method where these manually craffed feature vectors are replaced with word embedding using a convolutional neural network (CNN). Word embedding is then intensively

used as input of many NER systems [Yao et al., 2015; Zhou et al., 2017; Nguyen et al., 2016] in various domains. Word embedding can be either fixed or further fine-tuned during NER model training.

Character-level representation Several studies introduce a character-level representation learned by a neural model as an input of their systems. The advantages of this representation are basically: (i) to exploit sub-word-level information such as prefix and suffix; (ii) to handle out-of-vocabulary words. CNN-based model or RNN-based model (e.g. LSTM and Gated Recurrent Unit (GRU)) can be used to extract character-level representations. CharNER [Kuru et al., 2016] is a character-level tagger for language independent NER. CharNER uses LSTM to extract character-level representations and produces a tag distribution for each character instead of each word. Furthermore, Lample et al. [2016]; Ma and Hovy [2016] use a bidirectional LSTM to extract character-level representations of words. Each input vector is a concatenation of pre-trained word-level embedding and character-level representations. Chiu and Nichols [2016] present a hybrid bidirectional LSTM and a bidirectional CNN neural network architecture that help to exploit explicit character-level features such as prefixes and suffixes, which could be useful especially with rare words for which word embedding are poorly (or not) trained. Santos and Guimaraes [2015] introduce the neural character embedding in the NER task for English, and achieve the state-of-the-art. Jie and Lu [2019] propose a simple LSTM-CRF model for NER that takes the complete dependency trees. Anastasyev et al. [2018] explore ways to improve point-of-sale labeling using different types of auxiliary losses and different representations of words. They built their model based on Bi-LSTM layers, and showed that introducing word representations through their characters gives better results.

Hybrid representation In additional to word-level and character-level representations, some systems introduce additional information (e.g. lexical similarity, gazetteers, linguistic dependency and visual features) into the final representation of words. Enriching the input sequences of the neural network with accessible additional data can also help to have better results. Many systems prove that adding additional information improves the NER performance. Huang et al. [2015] use a BiLSTM-CRF with four types of features, namely spelling features, context features, word embedding, and gazetteer features. Their results show that combining multiple features improve the tagging accuracy. Wei et al. [2016] present a CRF system for NER of disease names, by employing multiple features, such as word embedding, POS tags, chunking, and word shape features (e.g., dictionary and morphological features). Lin et al. [2017] build their word representation by concatenating character-level representations, word-level representations, and

syntactical word representations (i.e., POS tags, dependency, word positions, head positions).

5.2.1.3.2 Context encoder The second step of NER deep learning is capturing the context dependencies using CNN, RNN or other network architectures. Collobert et al. [2011] propose a NER system using a CNN. The main problem with this system is that it does not take into account long-term dependencies between words, because it is based on a simple feed-forward neural network, and limits the use of context to a fixed-size window. To overcome this limitation, Collobert et al. [2011] propose a RNN deep learning algorithm for the sequence labeling task. Zhai et al. [2018] provide a comparison between NER systems using a CNN or RNN models. LSTM networks are a particular type of RNN that are designed to avoid this problem through the use of LSTM cells, which make it easy to learn about long-term dependencies [Gers et al., 1999]. Lample et al. [2016]; Huang et al. [2015] propose a more powerful neural network model that incorporates Bi-LSTM and CRF. The Bi-LSTM model takes into account the whole context, which enables it to effectively train a model with the flexible use of long-range context [Graves et al., 2013].

5.2.1.3.3 Tag decoder The final stage in a NER model is tag decoder. It takes context-dependent representations as input, and produces a sequence of tags corresponding to the input sequence. Multi-layer perceptron and a Softmax layer are used early as a tag decoder as numbers of NER models [Strubell et al., 2017; Li et al., 2017; Xu et al., 2017; Devlin et al., 2018; Cui and Zhang, 2019]. CRF represents the most common choice for tag decoder in many NER deep learning systems. Some are using CRF on top of the bidirectional LSTM layer [Huang et al., 2015; Zheng et al., 2017; Peters et al., 2018b; Lin et al., 2019]; others use CRF on top of the CNN layer [Collobert et al., 2011; Strubell et al., 2017; Yao et al., 2015].

When compared to the sate-of-the-art, our NER approach for the service request is mostly related to feature-inferring neural network systems. Character-based representations are very important in our use case. Since our data are user-generated, it is important to capture morphological and orthographic patterns.

5.2.2 Named entity linking

In this section, we focus on the related works for named entity linking. In the literature, many surveys are devoted to the named entity linking task [Al-Moslmi et al., 2020; Ling et al., 2015; Rao et al., 2013; Sevgili et al., 2020; Wu et al., 2018; Oliveira et al.; Shen et al., 2014]. These surveys present the different methods to perform the NEL task by following the general architecture of the NEL process which is composed of three steps:

(i) candidate entity generation which aims to retrieve a list of all possible entities in the knowledge base that may refer to each entity mention and filter out the irrelevant ones; (ii) candidate entity ranking aims to rank the list of candidate entities selected from the last step (i.e. candidate entity generation) and return the closest one for each entity mention; and (iii) NIL clustering or unlinkable mention prediction tries to deal with entities that do not have their corresponding entities in the knowledge base.

5.2.2.1 Candidate entity generation

There are three main methods to generate the entity candidate:

Dictionary based methods: These methods are based on the construction of a dictionary by using some features from Wikipedia such as entity pages, redirect pages, disambiguation pages, bold phrases from the first paragraphs, and hyperlinks. This type of dictionary provides information about name variations and represents a good way to generate candidate entities [Shen et al., 2014].

Surface form matching methods: These methods use the surface forms of mentions in the text to compose the candidates entities list.

Probability based methods: These methods compute the matching probability between the mentioned entity and the corresponding entity in the knowledge base to select the candidates entities. Usually a high value of probability implies that the entity in the knowledge base can be a selected candidate.

5.2.2.2 Candidate entity ranking

Three methods are presented in the literature to rank the candidate entity:

Similarity computation methods: These methods are based on the comparison of the similarity between the vector representation of each entity mention and the vector representation of the candidate entity in the knowledge graph using similarity measures (i.e cosine similarity, Jaccard similarity, etc.) To rank candidates entities, Bunescu and Pasca [2006] use the cosine similarity between the bag of word vector of the context of an entity mention and the bag of word vector of the Wikipedia page candidate entity. Then the target entity is represented by the entity with the maximum cosine similarity score.

Machine learning and deep learning methods: These methods use either a binary classification model such as SVM or Naive Bayes, or the deep learning methods to decide if a candidate entity is the target entity.

5.2.2.2.1 Graph methods: These methods are based on graph construction, in which the nodes represent the entities mentions and candidates entities, and the edges link each mention node with its candidates entities. After that, several techniques can be used to select the best candidate such as random walk for example.

5.2.2.3 NIL clustering

In this part, we give a brief overview of the three main approaches used in the literature to tackle with entities that have not been linked with mentions in the KB.

String matching methods: These methods compute the string similarity between entities to group them into clusters.

Hierarchical agglomerative clustering methods: These methods try to merge entities of each cluster until the distance between clusters is smaller than the threshold.

Graph methods: These methods are based on the construction of a semantic graph of entities before applying the hierarchical agglomerative clustering method.

Our NEL approach focuses on the first two steps of the NEL architecture and is based on the similarity score to choose the best concept for each extracted entity.

5.3 Our approach

5.3.1 Named entity recognition

Our aim in this step is to automatically analyze the textual descriptions of service requests and providers in order to extract the relevant entities. Those entities can be of different types. In general-purpose NER they are names of persons, organizations, places [Cardellino et al., 2017]. For the sourcing domain, we will consider names of skills, occupations, products (goods and services), and business activities.

The processing of these descriptions must address many challenges: (i) texts are generally short (50 words on average in our case); (ii) these texts are user-generated, and thus subject to typing errors; (iii) in some requests, users may describe their own products to contextualize their request, which would create confusion. This raises the issue of distinguishing between the user's real need and the general context of the sourcing request in its description.

To handle this type of textual data, our approach [Daoud et al., 2020] is inspired by [Lin et al., 2017] and is based on a Bi-LSTM-CRF architecture,

able to analyze textual descriptions of service providers and extract the relevant parts of the text that summarize a customer need or company offer. In addition to word embedding, we extract three other kinds of embedding for each word in the textual description: (i) a syntax based embedding; (ii) a character-level based embedding; and (iii) a position-based embedding. These three types of embedding are extracted with Bi-LSTM, and concatenated with a word embedding. As there is no large enough corpus available that specifically fits our use case to train a word embedding model, we resort to pre-trained word vectors for French, learned using fastText. The vector representation resulting from the concatenation of the four embeddings is given as input to our main Bi-LSTM-CRF model.

5.3.1.1 Syntactic word representations (SWR)

The manual analysis of our data shows that syntax plays an important role to locate the user's need in a description. For example, many service requests are using specific verbs like "rechercher"/"chercher" (to look for), or "souhaiter" (to wish) before explaining their needs. Therefore, recognizing the sentence object would help the model to recognize the customer's need. We then trained part-of-speech (POS) embedding, using a Bi-LSTM Model.

5.3.1.2 Character-level word representations (CLWR)

This representation is used to represent rare words or words with spelling errors, which cannot be captured with word embedding, since fastText embedding is limited to the vocabulary on the training corpus. For every word, we use a Bi-LSTM that takes as input the sequence of the characters of the words, and returns the vector of the last hidden states. We consider this vector as a character-level based representation.

5.3.1.3 Position representation (PR)

Based on manual analysis of our data, it turns out that usually the main subject is mentioned at the beginning of the text. Hence, we used this embedding type to push the model to understand that it is highly likely that the words at the beginning of the text are relevant information. We use a Bi-LSTM model to extract this type of embedding.

All these types of representations are concatenated and used as input of the main Bi-LSTM-CRF bloc as shown in Figure 5.1.

5.3.2 Named entity linking with the sourcing vocabulary

In order to link the named entities extracted from the descriptions of service requests and providers with the concepts in the sourcing vocabulary, we defined a similarity measure between an entity and a concept, and we link

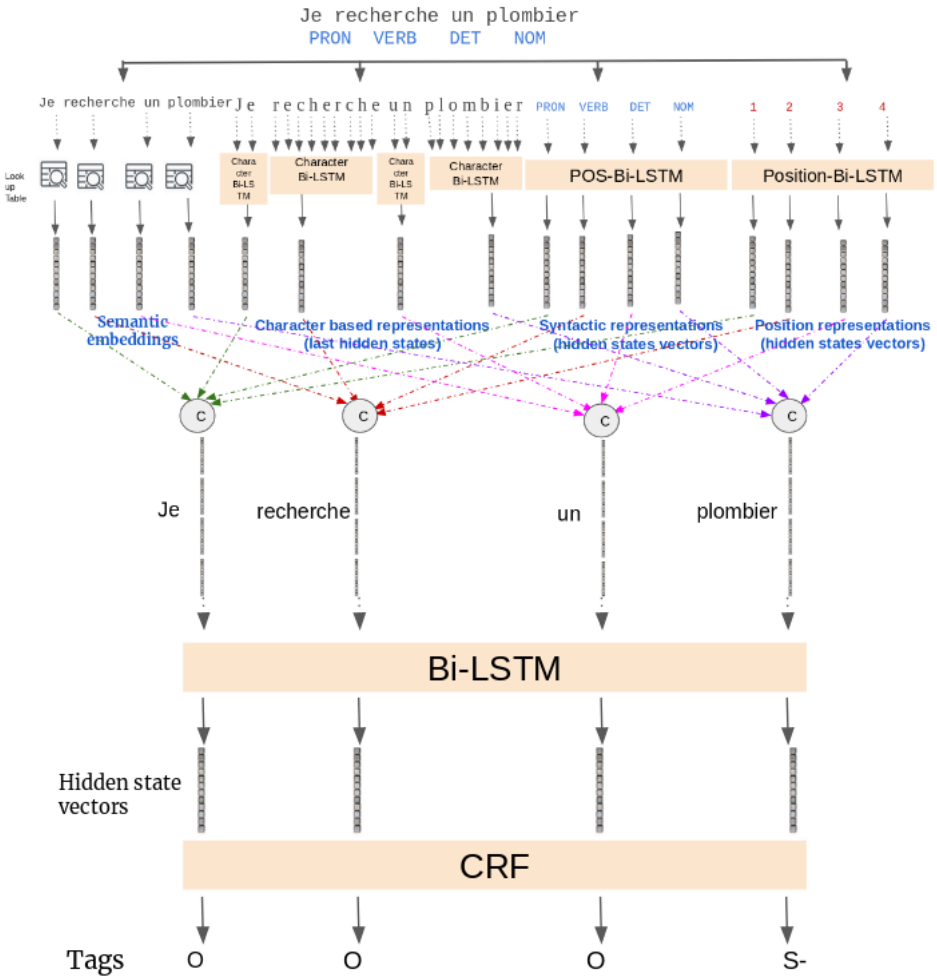


Figure 5.1: Model architecture (Embedding extraction + Main Bi-LSTM-CRF).

an entity to the closest concept in the vocabulary. We mention that in this step we do not use all the repositories building up the *Silex* vocabulary, we used ESCO, ROME, NAF, UNSPSC and CPF. So, we ignore the *silex* internal repository, cigref as well as kompass. This decision is based on the fact that the selected ontologies are more structured and cover all domains. We first represent each extracted entity and each concept of the sourcing vocabulary by an embedding vector which is computed as the average of the word embedding vectors of all the words participating in the entity or label of concept and occurring in the dictionary. Here again, in the absence of a large corpus available to train a word embedding model for our use case, we use pre-trained word vectors for French learned using fastText.

The embedding vector for an entity or a (label of) concept x is thus

computed as

$$V(x) = \frac{1}{n} \sum_{i=1}^{n} w_i, \tag{5.1}$$

where n is the number of words of the dictionary occurring in x and $w_i \in \mathbb{R}^{300}$ denotes the word embedding vector of the ith word of x occurring in the dictionary. If a word of x does not belong to the dictionary, it is just ignored.

Then we define the similarity between an entity e extracted from a request or provider description and a concept c in the sourcing vocabulary as the cosine similarity between their embedding vectors $V(e)$ and $V(c)$:

$$sim(e, c) = \frac{V(e) \cdot V(c)}{||V(e)|| \cdot ||V(c)||}. \tag{5.2}$$

Finally, we link each entity with the most similar concept in the vocabulary O:

$$linked_0(e, c) \iff sim(e, c) = \max_{c_i \in O} sim(e, c_i). \tag{5.3}$$

5.4 Experiments and results

5.4.1 Dataset and protocol

Our dataset for NER is composed of 883 descriptions of service requests distributed as follows: (i) 594 service request descriptions in the train set, (ii) 198 service request descriptions in the development set, and (iii) 90 service request descriptions in the test set. Data are annotated by sourcing experts at *Silex* according to the BIOES format which stands for Beginning (B-) to mark the beginning of an entity, Inside (I-) to mark the inside of an entity, Outside (O) to mark a token outside all of entities, End(E-) to mark the end of an entity, and Single (S-) to mark a single entity. Table 5.1 shows an example annotation using the BIOES format.

We conducted four experiments to compare the performance of four kinds of models:

- I: Bi-LSTM model with only word embedding and logistic regression classification model
- II: Bi-LSTM model with only word embedding and CRF classification model
- III: Bi-LSTM-CRF model with word embedding, character-based representations, and Bi-LSTM position based representation.
- IV: Bi-LSTM-CRF model with word embedding, character-based representations, Bi-LSTM position-based representation and syntactic word representations.

Word	POS	Label
Je	PRON	O
recherche	VERB	O
un	DET	O
plombier	NOUN	S-
La	DET	O
société	NOUN	O
BNB	NOUN	O
souhaite	VERB	O
créer	VERB	O
des	DET	O
supports	NOUN	B-
de	ADP	I-
communication	NOUN	E-
.	.	.
.	.	.

Table 5.1: Example of input training data.

5.4.2 Result and discussion

In order to evaluate the performance of the NER approach to extract knowledge from the textual description of service request, we used the precision, recall, F1 score to measure the match between the entities automatically extracted by the system and the entities manually produced by the experts. Let us note that even expert annotators find it sometimes hard to decide on the segment to annotate. In our evaluation, we do not consider the complete annotated entity, but rather words of the entity separately. For example, in the sentence *"I am looking for a **plumber** able to **repair a faucet Sprinkle**"*, we do not fully penalize the algorithm if it does not detect the complete **repair a faucet Sprinkle** segment. But we count words that it could detect in that segment. Indeed, if we suppose that the model detects only **repair a faucet**, this may be enough to understand the need's subject. We also ignore conjunctions, determinants and punctuation in the evaluation. Table 5.2 presents the results obtained in terms of precision, recall and F_1 score.

	Recall		Precision		F1	
Model	Dev	Test	Dev	Test	Dev	Test
I	58.88	61.31	77.02	76.61	66.74	68.11
II	64.03	66.61	75.21	76.66	69.17	71.29
III	63.06	66.61	75.62	**80.11**	68.77	**72.74**
IV	67.57	**70.20**	76.03	73.77	**71.55**	71.94

Table 5.2: Precision, Recall and F1-score.

Figure 5.2 and Figure 5.3 respectively show the evolution of the F1 score

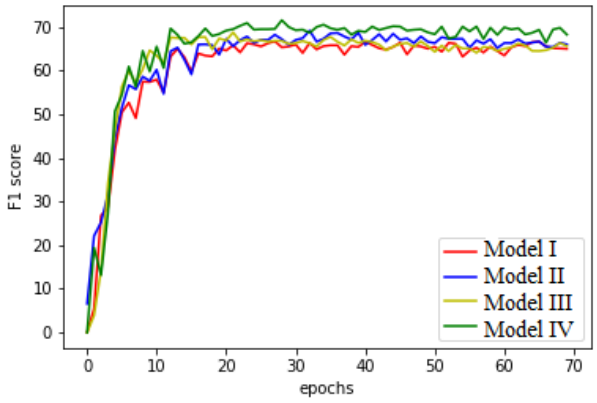


Figure 5.2: Evolution of score $F1_{Dev}$.

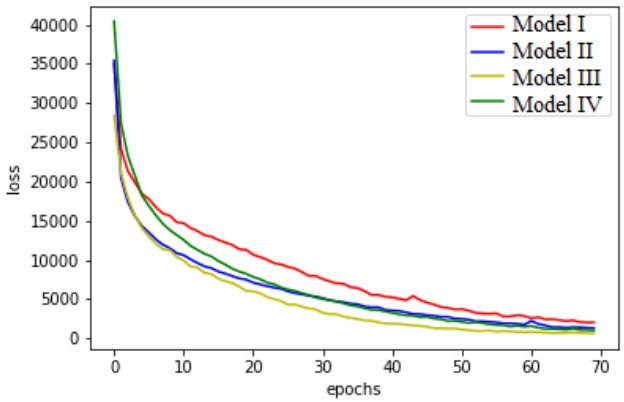


Figure 5.3: Evolution of the function. (*loss*)

and the evolution of the loss function across epochs.

One can see that model IV, which uses all types of features, is the best model across all epochs. Model I, which takes as word representations only word embedding with logistic regression for tagging, has the lowest score. Figure 5.3 shows that from epoch 30, the loss function continues to decrease without improving the F1 score. From this epoch, the model starts to overfit the training data.

Using syntactic information with POS tagging significantly improves Dev and Test recall, and balances well precision and recall (see Table 5.2). We also note that with this model, we were able to get a similar F1 scores in dev and test data (difference of 0.39%).

5.5 Conclusion

In this chapter, we detailed our knowledge extraction step by firstly proposing a method that relies on Bi-LSTM-CRF for sequence labeling to summarize service requests. We combined several types of features to represent every word in a sequence: (iii) character-level based embedding, (ii) syntax based embedding, and (iii) position embedding. These additional embeddings are extracted using Bi-LSTM and concatenated with word embedding. We showed that the syntax and position of words help to improve the quality of the knowledge extraction in our use case. Moreover, we showed that Bi-LSTM-CRF architectures for information extraction can provide value even in a small-data context. Secondly, we presented our NEL algorithm, which is based on textual context to measure the textual similarity between the context around the extracted entity and the concept of our sourcing vocabulary. We mentioned that we adopted a knowledge-based system approach to extract named entities from the textual description of providers. We didn't detail this work in this chapter as it is a simple lexical search of the different words of concepts in the textual descriptions of the providers.

Chapter 6

Recommender System

Contents

6.1 Introduction

Recommender Systems (RS) are a subclass of information filtering systems that seek to predict user preferences among a large selection of items [Ricci et al., 2011; Kumar and Reddy, 2014]. Nowadays, RS are primarily used in commercial applications to provide a personalized experience and suggest relevant items to users such as music, movies, books, trips, products, etc.

A strict word matching method between textual descriptions for the recommendation task would perform badly, as it does not take synonyms or polysemous words into account. To address that, word embedding models have been widely used to represent textual descriptions for the recommendation task in order to preserve the semantic and syntactic similarities between words. However, their main drawback is that they may lead to too generic a representation of texts, in the case where the available corpus is too small to train a specific word embedding model, and a pre-trained model is used instead, as it is the case in the context of the Silex platform.

In this chapter, we propose to combine a conceptual representation of texts to their representation based on word embedding to enhance the recommendation in the sourcing domain. Our main research question in this chapter is: Can the integration of domain knowledge enhance the performance of a RS in our use case in the sourcing domain? We focus on the following sub-questions:

- What is the best way to integrate domain knowledge into the representation of service requests and providers in order to enhance the quality of recommendations?
- To what extent does the injection of domain knowledge improve the performance of the system?

The remainder of this chapter is organized as follows. Section 6.2 gives an overview of state of the art on RS. Section 6.3 presents our approach. Section 6.4 reports and discusses the results of our experiments in the sourcing domain. Section 6.5 concludes and provides an outline of future work.

6.2 Related Work

RS generate meaningful recommendations to users for items that might be interesting to them. Several surveys on RS have been written [Guo et al., 2020; Singh et al.; Dong et al., 2020]. We can distinguish four types of RS:

6.2.1 Content-based recommender

A content-based recommender [Lang, 1995] analyzes the content (i.e. set of attributes or metadata) of items liked by users in the past, and suggests items

with similar content [Yu, 1999]. The similarity score between the user profile and the item profile is calculated using various methods such as Bayesian networks [Park et al., 2006], neural networks [Hidasi et al., 2015], decision trees [Golbandi et al., 2011], and TF-IDF [Huang et al., 2011]. The top-scoring items are then recommended to the user [Cheung et al., 2003]. The quality of the content-based recommendation can be improved over the time, and it can produce recommendations even with special tastes. However, it suffers from many drawbacks: (i) a poor recommendation result when the system does not have sufficient information about the particular preference of a new user (lack of information problem). (ii) The content-based system is not able to recommend new items that are different from the user preference, which limits the recommendation scope, and hides other items that are potentially interesting to the user (overspecialization).

Many content-based recommender systems use word embedding to represent the features of the processed texts. Musto et al. [2015] present a comparison of three kinds of vector representations based on word embedding to recommend movies and books: Latent Semantic Indexing, Random Indexing and word2vec. Shin et al. [2014] use word2vec to compute a vector representation of tags and recommend Tumblr blogs to users. Ozsoy [2016] uses word2vec to recommend the next check-in location based on social networks. Finally, Elsafty et al. [2018] use word embedding to build a recommender system for job postings.

6.2.2 Collaborative filtering

Collaborative filtering [Shardanand and Maes, 1995] is the most successful approach. It is based on the feedback (rating) on items provided by similar users. There are two main techniques of collaborative filtering: (i) Memory-based techniques try to find neighbors for a user or an item by computing the similarity between the active user and the other users or by focusing on the similarities between items; and (ii) Model-based techniques build an inference model using different machine learning algorithms, such as Bayesian networks, clustering, Markov decision processes, Sparse factor analysis, and rule-based approaches.

The advantage of collaborative filtering is that it does not require massive amounts of data to yield good results, and the overspecialization problem can be resolved by finding similar users in the system. However, the recommendation of a new item or new user can be poor when the system does not have any interaction information (users, previous recommendations), a problem known as cold start. There are also other problems such as the sparsity problem and synonym problem.

6.2.3 Knowledge-based recommendation

Knowledge-based recommendation [Ameen, 2019] is about using ontologies to model knowledge about the user context, the item context and the domain, and to compute the similarity between items. Using knowledge-based recommendation, many problems of common RS are eliminated such as the cold start problem, having a very good accuracy and explainable recommendation. In contrast, the major limitation for this RS approach is the cost of knowledge modelling. Many recommender systems use domain knowledge to enhance the performance of their systems. In the e-learning domain, Zhuhadar et al. [2009] propose a hybrid recommender system based on a multi-model domain ontology to represent the learning materials and a rule-based recommendation; Yu et al. [2007] propose an ontology-based approach for content recommendation by representing knowledge about the content, domain and the user context into an ontology; Tarus et al. [2017] propose an ontology-based method to model and represent the domain knowledge about the learner and learning resources. In the field of economics, Cantador et al. [2008] propose a multilayered approach for a hybrid recommendation model, where the user interests are represented as concepts of domain ontologies, and a collaborative recommendation mechanism is applied based on the similarities between such content-based user profiles. Cui et al. [2014] propose an approach for the recommendations of new items based on an ontology-based similarity and Matrix Factorization to predict the missing value in the user-item matrix. The similarity between items is dependent on their properties, that is, common features tend to increase similarity and non-common ones tend to diminish it. Werner et al. [2013] present a system to recommend a set of economic articles based on a set of ontologies used to describe both articles and user profiles. Other works use embedding-based methods [Zhang et al., 2016, 2018; Ernst et al., 2014] to either build knowledge graphs to enrich the representation of items or build user-items graphs by introducing users into the graph which can directly model the user preference. The traditional path-based methods Zhao et al. [2017]; Shi et al. [2015, 2018]; Wang et al. [2019] build a user-item graph and enrich the user and/or item representation using the path connectivity. Finally, unified methods [Li et al., 2019; Sha et al., 2019] try to benefit from both the semantic embedding of knowledge graphs and semantic path patterns to fully exploit the information in the knowledge graph. This approach is based on the notion of embedding propagation [Wang et al., 2018].

6.2.4 Hybrid recommendation

Hybrid recommendation [Adomavicius and Tuzhilin, 2005] combines the three above, to take advantage of the benefits of each of them, and to overcome the limitations of using only one type of method.

In our use case, since we do not have a large enough historical corpus of service requests associated to the relevant recommended providers, and since we do not have any feedback on recommended providers, we adopt a knowledge-based approach combined with embedding methods to represent the description of service requests and providers. Our choice is motivated by the fact that knowledge-based recommendations are neither dependent on ratings, nor on information about a particular user to give recommendations. Our method is at the intersection of the content-based and knowledge-based approaches. The rationale behind our approach is to take advantage of the power of word embedding, which approximates a general semantic relation, and the power of domain knowledge, which models a more specific semantic relation, to provide high-quality recommendations.

6.3 Proposed Approach

We suppose that (i) a provider is relevant for a service request if the descriptions of the service request and the provider are semantically similar and (ii) the semantic similarity of the descriptions of service requests and providers is a function of the similarity of the domain named entities or concepts they are using.

6.3.1 Vector representation of service requests and providers

We aim to represent each service request or provider by a vector that summarizes the semantics of the entities extracted from its description. For each description of a service request or provider, we consider three alternative vector representations: (i) the average of the embedding vectors of the entities in the textual description; (ii) a bag of concepts representation; and (iii) a vector representation combining the two former ones.

6.3.1.1 Word Embedding of entities

The base vector representation $V_{Emb}(x)$ of a service request or provider x is the average of the embedding vectors of all the entities $e_i, i = 1, ..., n$ extracted from its description:

$$V_{Emb}(x) = \frac{1}{n} \sum_{i=1}^{n} V(e_i), \tag{6.1}$$

where $V(e_i)$ is the vector representation of entity e_i as defined in Equation 5.1.

6.3.1.2 Bag of concepts

Using the result of the above described entity linking process, we consider an alternative representation $V_{BoC}(x)$ of a service request or provider x based on the sourcing vocabulary S: the bag of the concepts (BoC) in S which the entities e_i extracted from x are linked to according to Equation 5.3:

$$V_{BoC}(x) = BoC_S(x) = (b_1, ..., b_m) \tag{6.2}$$

where m is the size of the sourcing vocabulary S and $b_i = 1$ if $\exists e \in x, linked_S(e, c_i)$, and $b_i = 0$ otherwise. Additionally, we considered enriching the BoC representation of a service request or provider, by considering not only the concepts linked to the entities it contains but also some neighbors in the vocabulary to the linked concepts. More precisely, we considered the parents of the concepts linked to the entities (skos:narrower relation) or those semantically close (skos:closeMatch relation). Formally, we define three alternative BoC representations:

$$V'_{BoC}(x) = BoC'_S(x) = (b_1, ..., b_m) \tag{6.3}$$

where m is the size of the sourcing vocabulary S and $b_i = 1$ if $\exists e \in x, linked_S(e, c_i)$ or $linked_S(e, c_j)$ with c_j skos:narrower c_i; and $b_i = 0$ otherwise.

$$V''_{BoC}(x) = BoC''_S(x) = (b_1, ..., b_m) \tag{6.4}$$

where m is the size of the sourcing vocabulary S and $b_i = 1$ if $\exists e \in x, linked_S(e, c_i)$ or $linked_S(e, c_j)$ with c_j skos:closeMatch c_i; and $b_i = 0$ otherwise.

$$V'''_{BoC}(x) = BoC'''_S(x) = (b_1, ..., b_m) \tag{6.5}$$

where m is the size of the sourcing vocabulary S and $b_i = 1$ if $\exists e \in x, linked_S(e, c_i)$ or $linked_S(e, c_j)$ with c_j skos:narrower c_i or c_j skos:closeMatch c_i; and $b_i = 0$ otherwise.

6.3.1.3 Combination of vector representations

We define a third type of vector representation of a service request or provider as the concatenation of the vector representations defined in Equation 6.1 and one of the BoC representations defined in Equations 6.2, 6.3, 6.4, and 6.5, respectively:

$$V_{Conc}(x) = V_{Emb}(x)^\frown V_{BoC}(x), \tag{6.6}$$
$$V'_{Conc}(x) = V_{Emb}(x)^\frown V'_{BoC}(x), \tag{6.7}$$
$$V''_{Conc}(x) = V_{Emb}(x)^\frown V''_{BoC}(x), \tag{6.8}$$
$$V'''_{Conc}(x) = V_{Emb}(x)^\frown V'''_{BoC}(x). \tag{6.9}$$

Here, the $\frown$ symbol denotes the vector concatenation operator.

6.3.2 Recommendation algorithm

We define two metrics to measure the similarity between a service provider p and a service request r. The first one is the cosine similarity between the vector representations of p and r:

$$sim_1(p, r) = \frac{V(p) \cdot V(r)}{||V(p)|| \cdot ||V(r||}, \tag{6.10}$$

where V stands for one of the representations defined in Equations 6.1 to 6.9.

The second metrics is

$$sim_2(p, r) = \begin{cases} 1, & \text{if } V(p) \cap V(r) \neq 0, \\ sim_1(p, r), & \text{otherwise,} \end{cases} \tag{6.11}$$

where sim_1 is then computed with the base vector representation V_1.

A service provider p is recommended for a service request r if $sim_1(p, r)$ or $sim_2(p, r)$ is greater than a given threshold, depending on the chosen similarity measure. An empirical study was conducted to choose the value of the threshold, making its value vary and computing the recall and precision measures.

6.4 Experiments and results

6.4.1 Dataset and protocol

We evaluate the performance of our recommendation approach on a dataset provided by sales experts at Silex. This dataset comprises 109 descriptions of service requests and the 649 providers which were manually selected and recommended for these requests. They are in various areas: Computing, Marketing and commercial, Human resources management, General services, Finance, Industrial services, etc.

To evaluate and compare the different recommendation approaches defined in Section 6.3, we consider two sets of annotations of this dataset: A is the set of annotations automatically performed by our NER approach; A' is the result of a manual cleaning of A that we performed, after realizing that, for some descriptions, automatic extraction is still far from perfect and can introduce noise in the process; this noise can consist of entities that are useless for the description of exceedingly long entities. For example, for the following service request description : "Cabinets en étude et évaluations environnementales, etudes d'impacts, environnementales, d'actualisation de plan massif incendie, natura 2000. Evaluations environnementales", the named entity extracted is "Cabinet en étude" which is too generic and can lead to an erroneous recommendation, because it would match any research firm in any domain, not just environmental evaluation.

Table 6.1: Experimental settings ON RS.

Experiment	Vector representation	Similarity measure
`Emb`	V_{Emb}	sim_1
`BoC`	V_{BoC}	sim_1
`BoC′`	V'_{BoC}	sim_1
`BoC″`	V''_{BoC}	sim_1
`Boc‴`	V'''_{BoC}	sim_1
`Conc`	V_{Conc}	sim_1
`Conc′`	V'_{Conc}	sim_1
`Conc″`	V''_{Conc}	sim_1
`Conc‴`	V'''_{Conc}	sim_1
`BoC→Emb`	V_{BoC} and V_{Emb}	sim_2
`Conc→Emb`	V_{Conc} and V_{Emb}	sim_2

To decide on the optimal vector representation and algorithm to recommend service providers, we conducted eleven experiments whose settings are depicted in Table 6.1, the baseline being experiment `Emb`.

In order to evaluate the performance of the proposed settings, and therefore the interest of injecting domain knowledge into vector representations, we used the precision, recall, F1 score and "precision at N" metrics to measure the match between the recommendations automatically produced by the system and the recommendations manually produced by the experts. We conducted the parameter learning (i.e threshold) through 5-fold cross-validation.

6.4.2 Results and discussion

Table 6.2 and 6.3 presents the performance of our system for each tested setting in terms of precision, recall and F1 score with datasets A and A' respectively.

In order to evaluate to what extent the performance of our proposed approach depends on the method adopted to automatically annotate the descriptions of the service requests and providers, we conducted some additional experiments in which we used DBpedia spotlight[1] for NER. Table 6.4 presents the results obtained in terms of precision, recall and F1 score on dataset A when named entities are extracted with DBpedia spotlight [Mendes et al., 2011; Daiber et al., 2013].

Figures 6.1a and 6.1b present the performance of our system for each tested setting in terms of P@N on dataset A and dataset A' respectively.

Let us first discuss the results on dataset A. The best results were achieved with the `Conc` method, where the recommendation is based on a vector representation concatenating word embedding and BoC, and the

[1] https://www.dbpedia-spotlight.org/

Table 6.2: Evaluation of the proposed experimental settings with dataset A.

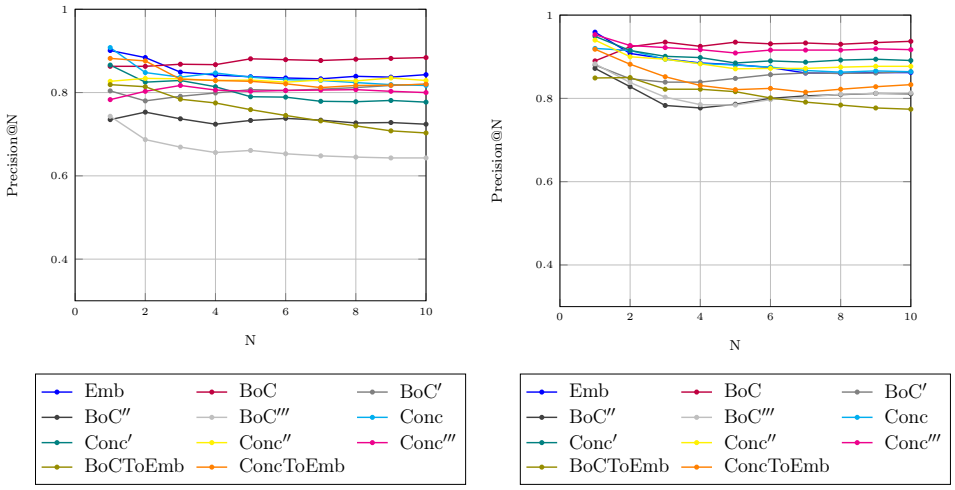
Experiment	Threshold	Precision	Recall	F1
Emb	0.76	0.848	0.397	0.530
BoC	0.239	**0.887**	0.137	0.235
BoC′	0.16	0.852	0.247	0.378
BoC″	0.15	0.728	0.187	0.296
BoC‴	0.069	0.697	0.287	0.402
Conc	0.58	0.875	**0.612**	**0.717**
Conc′	0.55	0.793	0.457	0.575
Conc ″	0.58	0.791	0.231	0.365
Conc ‴	0.52	0.797	0.258	0.389
BoC→Emb	0.70	0.600	0.580	0.579
Conc→Emb	0.24;0.76	0.83	0.429	0.558

Table 6.3: Evaluation of the proposed experimental settings with dataset A'.

Experiment	Threshold	Precision	Recall	F1
Emb	0.73	0.877	0.562	0.678
BoC	0.01	0.909	0.338	0.487
BoC′	0.16	0.890	0.456	0.596
BoC″	0.13	0.819	0.414	0.541
BoC‴	0.10	0.841	0.497	0.618
Conc	0.58	0.875	0.612	0.717
Conc ′	0.55	0.912	0.540	0.674
Conc ″	0.55	0.883	0.420	0.568
Conc ‴	0.52	**0.920**	0.430	0.584
BoC→Emb	0.7	0.72	**0.694**	**0.768**
Conc→Emb	0.05;0.73	0.860	0.627	0.719

Table 6.4: Evaluation of the proposed experimental settings with dataset A, using DBpedia spotlight for NER.

Experiment	Threshold	Precision	Recall	F1
Emb	0.79	0.346	0.169	0.219
BoC	0.11	**0.418**	**0.313**	**0.357**
Conc	0.39	0.401	0.193	0.260



(a) Dataset A. (b) Dataset A'.

Figure 6.1: Precision@N.

cosine similarity measure. The second best results were achieved with the `BoC→Emb` method which combines a BoC representation and word embedding. Unsurprisingly, considering the BoC representation increases the precision and F1 measures, even though at the expense of the recall score.

With dataset A', the best results based on the precision measure were achieved with the `Conc'''` method, where the recommendation is based on a vector representation concatenating word embedding and BoC with subsumption and skos:closeMatch relations. The best results in terms of F1 score were obtained using the `BoC→Emb` method which combines a BoC representation and word embedding. Using this cleaned dataset, the performance of all methods increases. This emphasizes the fact that all methods are very sensitive to the quality of entity linking.

All other methods, where a word embedding representation is enriched with domain knowledge, obtain a better precision measure than the baseline method using word embedding alone. Focusing on the precision@N results, with both datasets, we can conclude that injecting domain knowledge is highly beneficial to our RS. Although a BoC representation performs very well from the 2 top-ranking items on dataset A, we can observe that its performance keeps steady on both datasets up to the tenth item.

All in all, it appears that injecting domain knowledge into the vector representations is all the more beneficial, the greater the quality of the annotations. Also, enriching the conceptual representations by considering the subsumption relation and skos:closeMatch relation clearly gives better results. Finally, the comparison with the experiments using DBpedia spotlight for NER confirms that the introduction of domain knowledge in the recommendation process is beneficial and helps enhance the performance of the system

even when one cannot rely on a high-quality NER method.

6.5 Conclusion

In this chapter, we proposed a sourcing recommender system approach that exploits knowledge extracted from textual descriptions of providers and service requests to automatically suggest the best providers. In this work, we study the benefits of using ontological knowledge to improve our recommender process. We focus especially on the vector representation of the descriptions by evaluating the performance of the system using word embedding or injecting domain knowledge into the representation. We reported the results of our experiments on the *Silex* dataset. These experiments show that recommendation accuracy can be greatly improved through the injection of domain knowledge and especially the bag of concept representation in the recommendation process.

Chapter 7

Conclusion

Contents

This thesis falls within the framework of the RS domain. RS have become a crucial necessity for the companies and do not stop evolving these last years. In the context of *Silex*, RS is meant to be a specific type of decision support system that guides either directly the user or the *Silex* sale's staff to find the best providers for a service request. *Silex* collects data both by allowing users to provide a textual description of their professional activities, and by collecting the open data on the Web. As a result, the *Silex* dataset gathers data from all domains (i.e. computing, general service, marketing, Humans resources ...).

Motivated by the concerns of *Silex*, we addressed in this thesis the issue of defining a RS based on domain knowledge. Our approach relies on the following four bricks: (i) building a sourcing domain vocabulary, (ii) defining an ontology alignment approach to integrate various domain metadata repositories, (iii) defining a NER approach and a NEL approach to extract and link entities from the textual description of service requests and providers, and finally (iv) proposing a RS using conceptual representations to suggest the best suited service providers to answer a service request.

In this chapter, we summarize the work conducted in this thesis and we highlight our contributions. We report the limitations of our work and we discuss several future directions.

7.1 Summary of the thesis

The first contribution of this thesis is the construction of a vocabulary for the sourcing domain. This vocabulary has been used to semantically annotate text descriptions of service requests and providers based on four knowledge types: (i) skills, (ii) occupations, (iii) products, and (iv) business activities, in order to automatically generate high-quality recommendations of service providers. This vocabulary has been built using the NeON methodology by imagining usage scenarios and defining some competency questions. A top-down approach has been applied to reuse metadata repositories such as ESCO, ROME, NAF, UNSPSC and FCP. A bottom-up approach has been also applied to build an ontology from the company's internal data. Our sourcing domain vocabulary contains 125,488 concepts.

The second contribution of this thesis is the proposal of a novel approach to ontology alignment. The motivation of this work is to automatically align all metadata repositories used in the sourcing vocabulary. Our approach relies on the use of word embedding and measuring the spread clusters of labels to discover the relationship between entities. We tested our approach on multiple datasets: (i) the OAEI conference complex alignment benchmark, (ii) the real-world use case encountered by the *Silex* company, and (iii) the real-world use case encountered by the ONISEP. The experimental results show that the combination of word embedding and a measure of dispersion

of the clusters of labels, which we call the radius measure, makes it possible to determine, with good accuracy, not only equivalence relations, but also hierarchical relations between entities.

The third contribution of this thesis is the proposal of a knowledge extraction approach by presenting a NER approach based on a BiLSTM-CRF model, which combines four features types (i.e. word embedding, character-level based embedding, syntax based embedding, and position embedding), and a NEL algorithm based on computing the semantic similarity between the extracted entities and the concept of the sourcing vocabulary using word embedding. The experimental results relevant to the NER approach show that the combination of these four type of features helps to improve the quality of the knowledge extraction in the *Silex* use case.

The final contribution of this thesis addresses the proposal of a sourcing RS based on domain knowledge. We focused on the vector representation of the descriptions. Our proposal is to study the benefits of using knowledge, and more specifically, the 'bag of concepts' representation to enhance a RS in the sourcing domain. We tested our approach in a real-world case study provided by the *Silex* company. The experimental results show that injecting knowledge in the recommendation process outperforms word embedding approaches.

7.2 Technological transfer

As a result of the work carried out for this thesis, *Silex* now owns an aligned sourcing vocabulary. The sourcing vocabulary was integrated into the platform and was used to extract knowledge from the textual description of providers. This point is only briefly introduced in Chapter 5, because we have not evaluated the performance of this method yet. The NER approach applied to the service requests was integrated into the *Silex* platform.

Currently, usability tests for our RS approach are in progress and are based on a specific keyword set provided by the *Silex* product team to automatically recommend providers covering the whole provider database of *Silex*. We can also imagine that these tests can be useful not only for the integration of our approach in the platform, but also to know if our approach is sensitive to different business activities.

7.3 Limitations and perspectives of the proposed approach

The work presented in this thesis explores a range of techniques to automatically suggest the providers that best suit a service request. For some of them, there are several directions of improvements. In this section, we present some possible improvements and perspectives we aim at undertaking

Table 7.1: Summary of our work perspectives.

Steps	Roadmap	
	Terms	Perspectives
Construction of the sourcing vocabulary	Short	- Test other language detectors.
	Medium	- Automatically label hierarchical clusters.
	Long	- Add other multilingual metadata. - Evolution of sourcing knowledge in the context of each country.
Ontology alignment	Short	- Use a contextual embedding.
	Medium	- Evaluation on the entire *Silex* dataset or other datasets.
	Long	- Deal with leaf nodes.
NER and linking	Short	- Evaluate our NEL algorithm. - Use a contextual embedding.
	Medium	- Deal with unlinkable entities.
	Long	- Train our embedding model.
Recommender system	Short	- Integrate our work within the *Silex* platform.
	Medium	- Exploit additional metadata.
	Long	- Data visualization.

to overcome some of the limits identified in our work and to open follow-up research directions. Table 7.1 summarizes the perspectives of this section.

7.3.1 Construction of the sourcing vocabulary

The first improvement concerns the construction of the sourcing vocabulary. The actual vocabulary comprises a big number of concepts, but we regret that we do not use the richness of the internal vocabulary of *Silex* in an efficient way. We have started a vocabulary reconstruction work using a clustering approach, but what is currently lacking is a principled way to assign labels to the formed clusters to be able to align them with the the current vocabulary, and to give *Silex* the chance to have its own specific vocabulary. As a perspective, we aim at automatically labeling hierarchical clusters. A starting point will be the work of Treeratpituk and Callan [2006], who suppose that by comparing the word distribution of the hierarchy, it should be possible to assign appropriate labels to each cluster in the hierarchy. They propose an algorithm exploiting information on the cluster and the parent cluster and corpus statistics to assign a label to each cluster.

Another potential direction is to support *Silex* in its internationalisation phase, and that by improving the sourcing vocabulary with other multilingual metadata repositories. This is a key feature for *Silex* to open up to the inter-

national market. In the current version of our vocabulary, the most complete metadata that supports several European and non-European languages is ESCO, followed by UNSPSC. Adding the USA Occupational Information Network (O*NET),[1] the China Industry Classification System[2] and other industrial taxonomies can be a good track. We also encountered a small problem with UNSPSC: in the Excel file of the UNSPSC repository, the synonyms column provides the title synonyms of the concept in all languages separated with comma. We used the *langdetect* Python package to detect and automatically extract only the French language appellation from the synonyms columns. However, the extraction is not reliable enough, so we will have to think about another solution to have a good multilingual vocabulary. It will be good to find and test another language detector such as *textblob*[3] or *langrid* library [4] to extract more multilingual labels from UNSPSC repositories.

Another ambitious direction is to propose a continuous ontology updating solution that takes into account the permanent evolution of sourcing knowledge in the context of each country. We know that an ontology must to be modified over a period of time in order to reflect changes in the real word: a new concept might emerge, or a new concept might be introduced as the combination of two or more existing concepts (or conversely might be split into several concepts) Wardhana et al. [2018], etc. Therefore, capturing or even predicting these different changes is a real need in the field of sourcing.

7.3.2 Ontology alignment

The second improvement path concerns ontology alignment. The current algorithm presents the disadvantage of not being able to define the type of the relationship between two leaf nodes of ontologies. This is due to the fact that we use the notion of cluster, which is constructed in a top-down way in the ontology to deduce the type of relationship. To overcome the limitation of our ontology alignment algorithm on the leaf nodes, a potential solution would be to simply manipulate leaves differently from other nodes. Another improvement path would be to better capture the context of words using a contextual embedding. A now obvious perspective is to use BERT [Devlin et al., 2018] or CamemBERT [Martin et al., 2019] to represent each concept as a contextual embedding. On another note, it would also be useful to extend the evaluation of our system to the entire dataset or other datasets, and to empirically determine the optimal threshold value for the radius difference to be used in our alignment algorithm.

[1] https://www.onetonline.org/
[2] https://www.china-data-online.com/info/hyfl.asp
[3] https://textblob.readthedocs.io/en/dev/
[4] https://langrid.org/

7.3.3 Named entity recognition and linking

As far as the Named entity recognition and linking is concerned, several improvement paths could be explored. To begin with, it might be interesting to improve the NER method. We discussed before the fact that, for some descriptions, the extraction is not yet perfect (extraction too generic or too long). A possible path to improve our NER algorithm is to use again a contextual embedding to extract entities or ideally to train our embedding model on our data in order to better extract the relevant entities. Concerning the NEL algorithm, we used the same principle for entity linking as for ontology alignment, i.e. to identify links between extracted entities and concepts, or links between two concepts. As a consequence, at first we considered that the evaluation of our approach for ontology alignment was enough. It is time now to evaluate it. The improvement of our EL algorithm first goes through evaluating the results of our approach (and not only on ontology alignment). And just like for ontology alignment, a possible improvement path is to use a contextual embedding.

7.3.4 Recommender system

The most prominent limitation of our recommender system approach is the absence of link between entities and concepts. This may be due to a false extraction produced by the NER or to the inability of the EL algorithm to produce the link with ontology concepts or simply to the absence of this concept in our ontology. The bulk of this limitation can be resolved by improving the previous steps of our approach. A promising direction is to exploit additional metadata on service requestors and providers to improve their matching, like location, number of employees, sales turnover. In addition, the inclusion of historical user data constitutes an interesting extension of our work. Another ambitious direction will be to study the manner to present the results of the recommendation to users. This line of research is not yet addressed for at *Silex*. We think that from the user's point of view, it will be more interesting to present the results of recommendations in the form of a graph whose nodes are the providers, the service requestors as well as the ontology concepts, and whose edges are the different links between these three types of nodes. We can present these different types of nodes with different colors and vary the size of the nodes based on the number of concepts in common between a provider node and a service request node. This will naturally highlight the best recommendation results to the users. Finally, our RS should be integrated within the *Silex* platform and usability tests should be conducted.

In conclusion, it is true that the path of ontology engineering seems a long and difficult one, especially when applied in industry, but it always proves its performance and effectiveness.

Bibliography

Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6): 734–749, 2005.

Tareq Al-Moslmi, Marc Gallofré Ocaña, Andreas L Opdahl, and Csaba Veres. Named entity extraction for knowledge graphs: A literature overview. *IEEE Access*, 8:32862–32881, 2020.

Faisal Alshargi, Saeedeh Shekarpour, Tommaso Soru, and Amit Sheth. Concept2vec: Metrics for evaluating quality of embeddings for ontological concepts. *arXiv preprint arXiv:1803.04488*, 2018a.

Faisal Alshargi, Saeedeh Shekarpour, Tommaso Soru, and Amit P Sheth. Metrics for evaluating quality of embeddings for ontological concepts. 2018b.

Ayesha Ameen. Knowledge based recommendation system in semantic web -a survey. *International Journal of Computer Applications Volume 182 –No. 43*, 2019.

Fouzia Amourache, Zizette Boufaïda, and Leila Yahiaoui. Construction d'une ontologie basée compétence pour l'annotation des cvs/offres d'emploi. In *10th Conference on Software Engineering and Artificial Intelligence (MCSEAI), Maghrebian Conference on Information Technologies (28-30 april)*, pages 1–7, 2008.

Daniil Anastasyev, Ilya Gusev, and Eugene Indenbom. Improving part-of-speech tagging via multi-task learning and character-level word representations. *arXiv preprint arXiv:1807.00818*, 2018.

Grigoris Antoniou and Frank Van Harmelen. *A semantic web primer*. MIT press, 2004.

Fatima Ardjani, Djelloul Bouchiha, and Mimoun Malki. Ontology-alignment techniques: Survey and analysis. *International Journal of Modern Education & Computer Science*, 7(11), 2015.

Hisham Assal, John Seng, Franz Kurfess, Emily Schwarz, and Kym Pohl. Semantically-enhanced information extraction. In *2011 Aerospace Conference*, pages 1–14. IEEE, 2011.

David Aumueller, Hong-Hai Do, Sabine Massmann, and Erhard Rahm. Schema and ontology matching with coma++. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 906–908, 2005.

Mariette Awad and Rahul Khanna. *Efficient learning machines: theories, concepts, and applications for engineers and system designers*. Springer Nature, 2015.

John A Bateman. Kpml: the komet-penman (multilingual) development environment. Technical report, Technical report, GMD, Darmstadt. Release 0.8, 1995.

Pavel Berkhin. A survey of clustering data mining techniques. In *Grouping multidimensional data*, pages 25–71. Springer, 2006.

Amaia Bernaras. Building and reusing ontologies for electrical network applications. *Proc. of ECAI 96, 1996*, pages 298–302, 1996.

Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific american*, 284(5):34–43, 2001.

Daniel M Bikel, Scott Miller, Richard Schwartz, and Ralph Weischedel. Nymble: a high-performance learning name-finder. *arXiv preprint cmp-lg/9803003*, 1998.

Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data: The story so far. In *Semantic services, interoperability and web applications: emerging concepts*, pages 205–227. IGI Global, 2011.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795, 2013.

Paolo Bouquet, Luciano Serafini, Stefano Zanobini, and Simone Sceffer. Bootstrapping semantics on the web: meaning elicitation from schemas. In *Proceedings of the 15th international conference on World Wide Web*, pages 505–512, 2006.

Razvan Bunescu and Marius Pasca. Using encyclopedic knowledge for named entity disambiguation. 2006.

Robin Burke. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4):331–370, 2002.

Eleonora Cagli, Cécile Dumas, and Emmanuel Prouff. Convolutional neural networks with data augmentation against jitter-based countermeasures. In *International Conference on Cryptographic Hardware and Embedded Systems*, pages 45–68. Springer, 2017.

Horacio Camacho and Abdellah Salhi. A string metric based on a one-to-one greedy matching algorithm. *Research in Computer Science number*, 19: 171–182, 2006.

Iván Cantador, Alejandro Bellogín, and Pablo Castells. A multilayer ontology-based hybrid recommendation model. *Ai Communications*, 21(2-3):203–210, 2008.

Cristian Cardellino, Milagro Teruel, Laura Alonso Alemany, and Serena Villata. A low-cost, high-coverage legal named entity recognizer, classifier and linker. In *Proceedings of the 16th edition of the International Conference on Articial Intelligence and Law*, pages 9–18, 2017.

Michelle Cheatham and Pascal Hitzler. String similarity metrics for ontology alignment. In *International semantic web conference*, pages 294–309. Springer, 2013.

Kwok-Wai Cheung, James T Kwok, Martin H Law, and Kwok-Ching Tsui. Mining customer product ratings for personalized marketing. *Decision Support Systems*, 35(2):231–243, 2003.

Jason PC Chiu and Eric Nichols. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370, 2016.

William W Cohen, Pradeep Ravikumar, Stephen E Fienberg, et al. A comparison of string distance metrics for name-matching tasks. In *IIWeb*, volume 2003, pages 73–78, 2003.

Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167, 2008.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(Aug):2493–2537, 2011.

Oscar Corcho, Mariano Fernández-López, and Asunción Gómez-Pérez. Methodologies, tools and languages for building ontologies. where is their meeting point? *Data & knowledge engineering*, 46(1):41–64, 2003.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

Isabel F Cruz, Flavio Palandri Antonelli, and Cosmin Stroe. Agreementmaker: efficient matching for large real-world schemas and ontologies. *Proceedings of the VLDB Endowment*, 2(2):1586–1589, 2009.

Haomin Cui, Ming Zhu, and Shijia Yao. Ontology-based top-n recommendations on new items with matrix factorization. *Journal of Software*, 9(8): 2026–2032, 2014.

Leyang Cui and Yue Zhang. Hierarchically-refined label attention network for sequence labeling. *arXiv preprint arXiv:1908.08676*, 2019.

Dipak P Dabhi and Mihir R Patel. Extensive survey on hierarchical clustering methods in data mining. *International Research Journal of Engineering and Technology (IRJET)*, 3:659–665, 2016.

Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 9th International Conference on Semantic Systems (I-Semantics)*, 2013.

Hasnaa Daoud, Molka Tounsi Dhouib, Jerôme Rancati, Catherine Faron Zucker, and Andrea GB Tettamanzi. A hybrid bi-lstm-crf model for sequence labeling applied tothe sourcing domain. *6ème conférence sur les Applications Pratiques de l'Intelligence Artificielle APIA2020*, 2020.

J David. *AROMA: A method for the discovery of alignments between ontologies from association rules*. PhD thesis, Thèse d'informatique. Université de Nantes. Nantes (FR). URL:< http://tel ..., 2007.

Félix Hernández Del Olmo and Elena Gaudioso. Evaluation of recommender systems: A new approach. *Expert Systems with Applications*, 35(3):790–804, 2008.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Molka Dhouib, Catherine Faron Zucker, and Andrea Tettamanzi. Construction d'ontologie pour le domaine du sourcing. In *29es Journées Francophones d'Ingénierie des Connaissances, IC 2018*, pages 137–144, 2018.

Molka Tounsi Dhouib, Catherine Faron Zucker, and Andrea GB Tettamanzi. An ontology alignment approach combining word embedding and the radius measure. In *International Conference on Semantic Systems*, pages 191–197. Springer, 2019.

Molka Tounsi Dhouib, Catherine Faron, and Andrea Tettamanzi. Injection of knowledge in a sourcing recommender system. In *Web Intelligence*, 2020.

Hong-Hai Do and Erhard Rahm. Coma—a system for flexible combination of schema matching approaches. In *VLDB'02: Proceedings of the 28th International Conference on Very Large Databases*, pages 610–621. Elsevier, 2002.

AnHai Doan and Alon Y Halevy. Semantic integration research in the database community: A brief survey. *AI magazine*, 26(1):83–83, 2005.

Manqing Dong, Feng Yuan, Lina Yao, Xianzhi Wang, Xiwei Xu, and Liming Zhu. Trust in recommender systems: A deep learning perspective. *arXiv preprint arXiv:2004.03774*, 2020.

Sean R Eddy. Hidden markov models. *Current opinion in structural biology*, 6(3):361–365, 1996.

Marc Ehrig and Steffen Staab. Qom–quick ontology mapping. In *International Semantic Web Conference*, pages 683–697. Springer, 2004.

Lisa Ehrlinger and Wolfram Wöß. Towards a definition of knowledge graphs. *SEMANTiCS (Posters, Demos, SuCCESS)*, 48:1–4, 2016.

Ahmed Elsafty, Martin Riedl, and Chris Biemann. Document-based recommender system for job postings using dense representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 216–224, 2018.

Patrick Ernst, Cynthia Meng, Amy Siu, and Gerhard Weikum. Knowlife: a knowledge graph for health and life sciences. In *2014 IEEE 30th International Conference on Data Engineering*, pages 1254–1257. IEEE, 2014.

Jérôme Euzenat. Semantic precision and recall for ontology alignment evaluation. In *IJCAI*, volume 7, pages 348–353, 2007.

Jérôme Euzenat and Pavel Shvaiko. *Ontology matching*. Springer-Verlag, Heidelberg (DE), 2nd edition, 2013. ISBN 978-3-642-38720-3. URL http://book.ontologymatching.org.

Jérôme Euzenat and Petko Valtchev. Similarity-based ontology alignment in owl-lite. ecai, 2004.

Jérôme Euzenat, Pavel Shvaiko, et al. *Ontology matching*, volume 18. Springer, 2007.

Ronen Feldman and James Sanger. *The text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge university press, 2007.

Mariano Fernández-López and Asunción Gómez-Pérez. Overview and analysis of methodologies for building ontologies. *The knowledge engineering review*, 17(2):129, 2002.

Mariano Fernández-López, Asunción Gómez-Pérez, and Natalia Juristo. Methontology: from ontological art towards ontological engineering. 1997.

Fabien Gandon. *Graphes RDF et leur Manipulation pour la Gestion de Connaissances*. PhD thesis, 2008.

Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. 1999.

Fausto Giunchiglia, Pavel Shvaiko, and Mikalai Yatskevich. S-match: an algorithm and an implementation of semantic matching. In *European semantic web symposium*, pages 61–75. Springer, 2004.

Nadav Golbandi, Yehuda Koren, and Ronny Lempel. Adaptive bootstrapping of recommender systems using decision trees. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 595–604, 2011.

Wael H Gomaa, Aly A Fahmy, et al. A survey of text similarity approaches. *International Journal of Computer Applications*, 68(13):13–18, 2013.

Mari Carmen Gómez-Pérez, Asunción et Suárez-Figueroa. Neon methodology for building ontology networks: a scenario-based methodology. 2009.

Jorge Gracia and Eduardo Mena. Semantic heterogeneity issues on the web. *IEEE Internet Comput.*, 16(5):60–67, 2012. doi: 10.1109/MIC.2012.116. URL https://doi.org/10.1109/MIC.2012.116.

Michael Granitzer, Vedran Sabol, Kow Weng Onn, Dickson Lukose, and Klaus Tochtermann. Ontology alignment—a survey with focus on visually supported semi-automatic techniques. *Future Internet*, 2(3):238–258, 2010.

Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks*, 18(5-6):602–610, 2005.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE, 2013.

Ralph Grishman and Beth M Sundheim. Message understanding conference-6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*, 1996.

Dagmar Gromann and Thierry Declerck. Comparing pretrained multilingual word embeddings on an ontology alignment task. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

Thomas R Gruber. Toward principles for the design of ontologies used for knowledge sharing? *International journal of human-computer studies*, 43 (5-6):907–928, 1995.

Michael Grüninger and Mark S Fox. Methodology for the design and evaluation of ontologies. 1995.

Asela Gunawardana and Guy Shani. A survey of accuracy evaluation metrics of recommendation tasks. *Journal of Machine Learning Research*, 10(12), 2009.

Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, and Qing He. A survey on knowledge graph-based recommender systems. *arXiv preprint arXiv:2003.00911*, 2020.

Marios Hadjieleftheriou and Divesh Srivastava. Weighted set-based string similarity. *IEEE Data Eng. Bull.*, 33(1):25–36, 2010.

Uri Hanani, Bracha Shapira, and Peretz Shoval. Information filtering: Overview of issues, research and systems. *User modeling and user-adapted interaction*, 11(3):203–259, 2001.

Walid Hassen. Medley results for oaei 2012. In *Proceedings of the 7th International Conference on Ontology Matching-Volume 946*, pages 168–172. CEUR-WS. org, 2012.

Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28, 1998.

Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015.

Felix Hill, Roi Reichart, and Anna Korhonen. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695, 2015.

Cheng-Hui Huang, Jian Yin, and Fang Hou. A text similarity measurement combining word semantic information with tf-idf method. *Jisuanji Xuebao(Chinese Journal of Computers)*, 34(5):856–864, 2011.

Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

Rizwan Iqbal, Masrah Azrifah Azmi Murad, Aida Mustapha, Nurfadhlina Mohd Sharef, et al. An analysis of ontology engineering methodologies: A literature review. *Research journal of applied sciences, engineering and technology*, 6(16):2993–3000, 2013.

Abhyuday N Jagannatha and Hong Yu. Structured prediction models for rnn based sequence labeling in clinical text. In *Proceedings of the conference on empirical methods in natural language processing. conference on empirical methods in natural language processing*, volume 2016, page 856. NIH Public Access, 2016.

Yves R Jean-Mary, E Patrick Shironoshita, and Mansur R Kabuka. Ontology matching with semantic verification. *Journal of Web Semantics*, 7(3): 235–251, 2009.

Ningsheng Jian, Wei Hu, Gong Cheng, and Yuzhong Qu. Falcon-ao: Aligning ontologies with falcon. In *Proceedings of K-CAP Workshop on Integrating Ontologies*, pages 85–91, 2005.

Ridong Jiang, Rafael E Banchs, and Haizhou Li. Evaluating and combining name entity recognition systems. In *Proceedings of the Sixth Named Entity Workshop*, pages 21–27, 2016a.

Shangpu Jiang, Daniel Lowd, Sabin Kafle, and Dejing Dou. Ontology matching with knowledge rules. In *Transactions on Large-Scale Data- and Knowledge-Centered Systems XXVIII*, pages 75–95. Springer, 2016b.

Zhanming Jie and Wei Lu. Dependency-guided lstm-crf for named entity recognition. *arXiv preprint arXiv:1909.10148*, 2019.

Yannis Kalfoglou and Marco Schorlemmer. Ontology mapping: the state of the art. *The knowledge engineering review*, 18(1):1–31, 2003.

Prodromos Kolyvakis, Alexandros Kalousis, and Dimitris Kiritsis. Deepalignment: Unsupervised ontology matching with refined word vectors. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 787–798, 2018.

Michal Konkol and Miloslav Konopík. Crf-based czech named entity recognizer and consolidation of czech ner research. In *International conference on text, speech and dialogue*, pages 153–160. Springer, 2013.

Konstantinos I Kotis, George A Vouros, and Dimitris Spiliotopoulos. Ontology engineering methodologies for the evolution of living and reused ontologies: status, trends, findings and recommendations. *The Knowledge Engineering Review*, 35, 2020.

Markus Krötzsch. Ontologies for knowledge graphs? In *Description Logics*, 2017.

PN Vijaya Kumar and V Raghunatha Reddy. A survey on recommender systems (rss) and its applications. *International Journal of Innovative Research in Computer and Communication Engineering*, 2(8):5254–5260, 2014.

Onur Kuru, Ozan Arkan Can, and Deniz Yuret. Charner: Character-level named entity recognition. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 911–921, 2016.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*, 2016.

Ken Lang. Newsweeder: Learning to filter netnews. In *Machine Learning Proceedings 1995*, pages 331–339. Elsevier, 1995.

Juan J Lastra-Díaz, Josu Goikoetxea, Mohamed Ali Hadj Taieb, Ana García-Serrano, Mohamed Ben Aouicha, and Eneko Agirre. A reproducible survey on word embeddings and ontology-based methods for word similarity: linear combinations outperform the state of the art. *Engineering Applications of Artificial Intelligence*, 85:645–665, 2019.

Masupha Lerato, Omobayo A Esan, Ashley-Dejo Ebunoluwa, SM Ngwira, and Tranos Zuva. A survey of recommender system feedback techniques,

comparison and evaluation metrics. In *2015 International Conference on Computing, Communication and Security (ICCCS)*, pages 1–4. IEEE, 2015.

Michael Lesk. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on Systems documentation*, pages 24–26. Citeseer, 1986.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition. *arXiv preprint arXiv:1812.09449*, 2018.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

Juanzi Li, Jie Tang, Yi Li, and Qiong Luo. Rimom: A dynamic multistrategy ontology alignment framework. *IEEE Transactions on Knowledge and data Engineering*, 21(8):1218–1232, 2008.

Peng-Hsuan Li, Ruo-Ping Dong, Yu-Siang Wang, Ju-Chieh Chou, and Wei-Yun Ma. Leveraging linguistic structures for named entity recognition with bidirectional recursive neural networks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2664–2669, 2017.

Qianyu Li, Xiaoli Tang, Tengyun Wang, Haizhi Yang, and Hengjie Song. Unifying task-oriented knowledge graph learning and recommendation. *IEEE Access*, 7:115816–115828, 2019.

Bill Yuchen Lin, Frank F Xu, Zhiyi Luo, and Kenny Zhu. Multi-channel bilstm-crf model for emerging named entity recognition in social media. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 160–165, 2017.

Feiyu Lin and Kurt Sandkuhl. A survey of exploiting wordnet in ontology matching. In *IFIP International Conference on Artificial Intelligence in Theory and Practice*, pages 341–350. Springer, 2008.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Y Liu, and X Zhu. Learning entity and relation embeddings for knowledge graph completion, 2015. *Google Scholar Google Scholar Digital Library Digital Library.*

Yankai Lin, Zhiyuan Liu, Huanbo Luan, Maosong Sun, Siwei Rao, and Song Liu. Modeling relation paths for representation learning of knowledge bases. *arXiv preprint arXiv:1506.00379*, 2015.

Ying Lin, Liyuan Liu, Heng Ji, Dong Yu, and Jiawei Han. Reliability-aware dynamic feature composition for name tagging. In *Proceedings of the 57th*

Annual Meeting of the Association for Computational Linguistics, pages 165–174, 2019.

Greg Linden, Brent Smith, and Jeremy York. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, 7(1):76–80, 2003.

Xiao Ling, Sameer Singh, and Daniel S Weld. Design challenges for entity linking. *Transactions of the Association for Computational Linguistics*, 3: 315–328, 2015.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*, 2016.

Ian MacKenzie, Chris Meyer, and Steve Noble. How retailers can keep up with consumers. *McKinsey & Company*, 18, 2013.

Jayant Madhavan, Philip A Bernstein, and Erhard Rahm. Generic schema matching with cupid. In *vldb*, volume 1, pages 49–58, 2001.

Robert Malouf. Markov models for language-independent named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*, 2002.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte de la Clergerie, Djamé Seddah, and Benoît Sagot. Camembert: a tasty french language model. *arXiv preprint arXiv:1911.03894*, 2019.

Andrew McCallum and Wei Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. 2003.

Andrew McCallum, Dayne Freitag, and Fernando CN Pereira. Maximum entropy markov models for information extraction and segmentation. In *Icml*, volume 17, pages 591–598, 2000.

Paul McNamee and James Mayfield. Entity extraction without language-specific resources. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*, 2002.

Pablo N. Mendes, Max Jakob, Andres Garcia-Silva, and Christian Bizer. Dbpedia spotlight: Shedding light on the web of documents. In *Proceedings*

of the 7th International Conference on Semantic Systems (I-Semantics), 2011.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013a.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013b.

Tomas Mikolov, Kai Chen, Gregory S Corrado, and Jeffrey A Dean. Computing numeric representations of words in a high-dimensional space, May 19 2015. US Patent 9,037,464.

Majid Mohammadi, Amir Ahooye Atashin, Wout Hofman, and Yaohua Tan. Comparison of ontology alignment systems across single matching task via the mcnemar's test. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 12(4):51, 2018.

Naji F Mohammed and Nazlia Omar. Arabic named entity recognition using artificial neural network. *Journal of Computer Science*, 8(8):1285, 2012.

Alvaro E Monge, Charles Elkan, et al. The field matching problem: Algorithms and applications. In *Kdd*, volume 2, pages 267–270, 1996.

Cataldo Musto, Giovanni Semeraro, Marco De Gemmis, and Pasquale Lops. Word embedding techniques for content-based recommender systems: An empirical evaluation. In *Recsys posters*, 2015.

David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

DuyHoa Ngo and Zohra Bellahsene. Yam++: a multi-strategy based approach for ontology matching task. In *International Conference on Knowledge Engineering and Knowledge Management*, pages 421–425. Springer, 2012.

Nam Nguyen and Yunsong Guo. Comparisons of sequence labeling algorithms and extensions. In *Proceedings of the 24th international conference on Machine learning*, pages 681–688, 2007.

Thien Huu Nguyen, Avirup Sil, Georgiana Dinu, and Radu Florian. Toward mention detection robustness with recurrent neural networks. *arXiv preprint arXiv:1602.07749*, 2016.

Van Tien Nguyen, Christian Sallaberry, and Mauro Gaio. Mesure de la similarit\'e entre termes et labels de concepts ontologiques. *arXiv preprint arXiv:1307.6422*, 2013.

Sergei Nirenburg and Christine Defrise. Application-oriented computational semantics. *Computational Linguistics and Formal Semantics*, pages 223–256, 1992.

Ikechukwu Nkisi-Orji, Nirmalie Wiratunga, Stewart Massie, Kit-Ying Hui, and Rachel Heaven. Ontology alignment based on word embedding and random forest classification. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 557–572. Springer, 2018.

Natalya F Noy and Deborah L McGuinness. Développement d'une ontologie 101: Guide pour la création de votre première ontologie. *Université de Stanford, Stanford, CA, 94305. Traduit de l'anglais par Anila Angjeli, BnF, Bureau de normalisation document*, 2000.

Natalya Fridman Noy and Mark A Musen. Anchor-prompt: Using non-local context for semantic matching. In *OIS@ IJCAI*, 2001.

Peter Ochieng and Swaib Kyanda. Large-scale ontology matching: State-of-the-art analysis. *ACM Computing Surveys (CSUR)*, 51(4):75, 2018.

Italo L Oliveira, Renato Fileto, René Speck, Luís PF Garcia, Diego Moussallem, and Jens Lehmann. Towards holistic entity linking: Survey and directions. *Information Systems*, 95:101624.

Lorena Otero-Cerdeira, Francisco J Rodríguez-Martínez, and Alma Gómez-Rodríguez. Ontology matching: A literature review. *Expert Systems with Applications*, 42(2):949–971, 2015.

Makbule Gulcin Ozsoy. From word embeddings to item recommendation. *arXiv preprint arXiv:1601.01356*, 2016.

Han-Saem Park, Ji-Oh Yoo, and Sung-Bae Cho. A context-aware music recommendation system using fuzzy bayesian networks with utility theory. In *International conference on Fuzzy systems and knowledge discovery*, pages 970–979. Springer, 2006.

Daniel Parrochia and Pierre Neuville. *Taxinomie et réalité: vers une méta-classification*. ISTE Group, 2014.

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018a.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018b.

pôle emploi. Correspondance naf 2008-rome v3, June 2017. URL http://www.pole-emploi.org/files/live/sites/peorg/files/documents/Statistiques-et-analyses/Open-data/ROME/rome_correspondance_naf_juin.pdf.

Hanieh Poostchi, Ehsan Zare Borzeshi, and Massimo Piccardi. Bilstm-crf for persian named-entity recognition armanpersonercorpus: The first entity-annotated persian dataset. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

J. Ross Quinlan. Simplifying decision trees. *International journal of man-machine studies*, 27(3):221–234, 1987.

M Kuchaki Rafsanjani, Z Asghari Varzaneh, and N Emami Chukanlo. A survey of hierarchical clustering algorithms. *The Journal of Mathematics and Computer Science*, 5(3):229–240, 2012.

Erhard Rahm and Philip A Bernstein. A survey of approaches to automatic schema matching. *the VLDB Journal*, 10(4):334–350, 2001.

Delip Rao, Paul McNamee, and Mark Dredze. Entity linking: Finding extracted entities in a knowledge base. In *Multi-source, multilingual information extraction and summarization*, pages 93–115. Springer, 2013.

Payam Refaeilzadeh, Lei Tang, and Huan Liu. *Cross-Validation*, pages 532–538. Springer US, Boston, MA, 2009. ISBN 978-0-387-39940-9. doi: 10.1007/978-0-387-39940-9_565. URL https://doi.org/10.1007/978-0-387-39940-9_565.

Paul Resnick and Hal R Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997.

Francesco Ricci, Lior Rokach, and Bracha Shapira. Introduction to recommender systems handbook. In *Recommender systems handbook*, pages 1–35. Springer, 2011.

Petar Ristoski, Stefano Faralli, Simone Paolo Ponzetto, and Heiko Paulheim. Large-scale taxonomy induction using entity and word embeddings. In *Proceedings of the International Conference on Web Intelligence*, pages 81–87. ACM, 2017.

Dominique Ritze, Christian Meilicke, Ondřej Šváb-Zamazal, and Heiner Stuckenschmidt. A pattern-based ontology matching approach for detecting complex correspondences. In *ISWC Workshop on Ontology Matching, Chantilly (VA US)*, pages 25–36, 2009.

Dominique Ritze, Johanna Völker, Christian Meilicke, and Ondrej Sváb-Zamazal. Linguistic analysis for complex ontology matching. In *CEUR Workshop Proceedings*, volume 689, pages Paper–1. RWTH, 2010.

Magnus Sahlgren and Rickard Cöster. Using bag-of-concepts to improve the performance of support vector machines in text categorization. 2004.

Cicero Nogueira dos Santos and Victor Guimaraes. Boosting named entity recognition with neural character embeddings. *arXiv preprint arXiv:1505.05008*, 2015.

Daniela Schmidt, Rafael Basso, Cassia Trojahn, and Renata Vieira. Matching domain and top-level ontologies exploring word sense disambiguation and word embedding. In *Ontology Matching: OM-2018: Proceedings of the ISWC Workshop*, page 1, 2018.

Ozge Sevgili, Artem Shelmanov, Mikhail Arkhipov, Alexander Panchenko, and Chris Biemann. Neural entity linking: A survey of models based on deep learning. *arXiv preprint arXiv:2006.00575*, 2020.

Xiao Sha, Zhu Sun, and Jie Zhang. Attentive knowledge graph embedding for personalized recommendation. *arXiv preprint arXiv:1910.08288*, 2019.

Lipi Shah, Hetal Gaudani, and Prem Balani. Survey on recommendation system. *International Journal of Computer Applications*, 137(7):43–49, 2016.

Upendra Shardanand and Pattie Maes. Social information filtering: algorithms for automating "word of mouth". In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 210–217, 1995.

Wei Shen, Jianyong Wang, and Jiawei Han. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):443–460, 2014.

Chuan Shi, Zhiqiang Zhang, Ping Luo, Philip S Yu, Yading Yue, and Bin Wu. Semantic path based personalized recommendation on weighted heterogeneous information networks. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 453–462, 2015.

Chuan Shi, Binbin Hu, Wayne Xin Zhao, and S Yu Philip. Heterogeneous information network embedding for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 31(2):357–370, 2018.

Donghyuk Shin, Suleyman Cetintas, and Kuang-Chih Lee. Recommending tumblr blogs to follow with inductive matrix completion. In *RecSys Posters*, 2014.

Pavel Shvaiko and Jérôme Euzenat. A survey of schema-based matching approaches. In *Journal on data semantics IV*, pages 146–171. Springer, 2005.

Pavel Shvaiko and Jérôme Euzenat. Ontology matching: state of the art and future challenges. *IEEE Transactions on knowledge and data engineering*, 25(1):158–176, 2011.

Elena Simperl and Markus Luczak-Rösch. Collaborative ontology engineering: a survey. 2014.

Pradeep Kumar Singh, Pijush Kanti Dutta, Avick Kumar Dey Pramanik, and Prasenjit Choudhury. Recommender systems: An overview, research trends, and future directions.

Sonit Singh. Natural language processing for information extraction. *arXiv preprint arXiv:1807.02383*, 2018.

Amit Singhal et al. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, 24(4):35–43, 2001.

Steffen Staab, Rudi Studer, H-P Schnurr, and York Sure. Knowledge processes and ontologies. *IEEE Intelligent systems*, 16(1):26–34, 2001.

Bernd Stadlhofer, Peter Salhofer, and Augustin Durlacher. An overview of ontology engineering methodologies in the context of public administration. In *Proceedings of the 7th International Conference on Advances in Semantic Processing, IARIA, Porto, Portugal*, volume 29, pages 36–42, 2013.

Giorgos Stoilos, Giorgos Stamou, and Stefanos Kollias. A string metric for ontology alignment. In *International Semantic Web Conference*, pages 624–637. Springer, 2005.

Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. Fast and accurate entity recognition with iterated dilated convolutions. *arXiv preprint arXiv:1702.02098*, 2017.

Mari Carmen Suárez-Figueroa. *NeOn Methodology for building ontology networks: specification, scheduling and reuse*. PhD thesis, Informatica, 2010.

Charles Sutton, Andrew McCallum, et al. An introduction to conditional random fields. *Foundations and Trends® in Machine Learning*, 4(4): 267–373, 2012.

Bill Swartout, Ramesh Patil, Kevin Knight, and Tom Russ. Toward distributed use of large-scale ontologies. In *Proc. of the Tenth Workshop on Knowledge Acquisition for Knowledge-Based Systems*, pages 138–148, 1996.

Oscar Täckström. An evaluation of bag-of-concepts representations in automatic text classification. *Recall*, 2005.

John K Tarus, Zhendong Niu, and Abdallah Yousif. A hybrid knowledge-based recommender system for e-learning based on ontology and sequential pattern mining. *Future Generation Computer Systems*, 72:37–48, 2017.

Elodie Thieblin. Task-oriented complex alignments on conference organisation, 4 2019. URL https://figshare.com/articles/Complex_alignment_dataset_on_conference_organisation/4986368.

Elodie Thiéblin, Ollivier Haemmerlé, Nathalie Hernandez, and Cassia Trojahn. Un jeu de données d'évaluation de correspondances complexes entre ontologies. 2017.

Élodie Thiéblin, Ollivier Haemmerlé, Nathalie Hernandez, and Cassia Trojahn. Task-oriented complex ontology alignment: Two alignment evaluation sets. In *European Semantic Web Conference*, pages 655–670. Springer, 2018.

Pucktada Treeratpituk and Jamie Callan. Automatically labeling hierarchical clusters. In *Proceedings of the 2006 international conference on Digital government research*, pages 167–176, 2006.

Richard Tzong-Han Tsai, Shih-Hung Wu, Wen-Chi Chou, Yu-Chun Lin, Ding He, Jieh Hsiang, Ting-Yi Sung, and Wen-Lian Hsu. Various criteria in the evaluation of biomedical named entity recognition. *BMC bioinformatics*, 7 (1):92, 2006.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics, 2010.

Michael Uschold and Martin King. *Towards a methodology for building ontologies*. Citeseer, 1995.

Michael Uschold, Michael Gruninger, et al. Ontologies: Principles, methods and applications. *TECHNICAL REPORT-UNIVERSITY OF EDINBURGH ARTIFICIAL INTELLIGENCE APPLICATIONS INSTITUTE AIAI TR*, 1996.

Rafael Vieira and Kate Revoredo. Using word semantics on entity names for correspondence set generation. In *OM@ ISWC*, pages 223–224, 2017.

Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 417–426, 2018.

Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. Explainable reasoning over knowledge graphs for recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5329–5336, 2019.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Aaai*, volume 14, pages 1112–1119. Citeseer, 2014.

Helna Wardhana, Ahmad Ashari, and A Kartika. Review of ontology evolution process. *International Journal of Computer Applications*, 45:26–33, 2018.

Qikang Wei, Tao Chen, Ruifeng Xu, Yulan He, and Lin Gui. Disease named entity recognition by combining conditional random fields and bidirectional recurrent neural networks. *Database*, 2016, 2016.

David Werner, Christophe Cruz, and Christophe Nicolle. Ontology-based recommender system of economic articles. *arXiv preprint arXiv:1301.4781*, 2013.

Daya C Wimalasuriya and Dejing Dou. Ontology-based information extraction: An introduction and a survey of current approaches, 2010.

Gongqing Wu, Ying He, and Xuegang Hu. Entity linking: an issue to extract corresponding entity with knowledge base. *IEEE Access*, 6:6220–6231, 2018.

Han Xiao, Minlie Huang, Yu Hao, and Xiaoyan Zhu. Transg: A generative mixture model for knowledge graph embedding. *arXiv preprint arXiv:1509.05488*, 2015.

Mingbin Xu, Hui Jiang, and Sedtawut Watcharawittayakul. A local detection approach for named entity recognition and mention detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1237–1247, 2017.

Usha Yadav, Gagandeep Singh Narula, Neelam Duhan, and Vishal Jain. Ontology engineering and development aspects: a survey. *International Journal of Education and Management Engineering JEME*, 2016.

Vikas Yadav and Steven Bethard. A survey on recent advances in named entity recognition from deep learning models. *arXiv preprint arXiv:1910.11470*, 2019.

Xin-She Yang. *Introduction to Algorithms for Data Mining and Machine Learning*. Academic Press, 2019.

Lin Yao, Hong Liu, Yi Liu, Xinxin Li, and Muhammad Waqas Anwar. Biomedical named entity recognition based on deep neutral network. *Int. J. Hybrid Inf. Technol*, 8(8):279–288, 2015.

Philip S Yu. Data mining and personalization technologies. In *Proceedings. 6th International Conference on Advanced Systems for Advanced Applications*, pages 6–13. IEEE, 1999.

Zhiwen Yu, Yuichi Nakamura, Seiie Jang, Shoji Kajita, and Kenji Mase. Ontology-based semantic recommendation for context-aware e-learning. In *International Conference on Ubiquitous Intelligence and Computing*, pages 898–907. Springer, 2007.

Manuel Zacklad. Classification, thésaurus, ontologies, folksonomies: comparaisons du point de vue de la recherche ouverte d'information (roi). In *CAIS/ACSI 2007, 35e Congrès annuel de l'Association Canadienne des Sciences de l'Information. Partage de l'information dans un monde fragmenté: Franchir les frontières, sous la dir. de C. Arsenault et K. Dalkir. Montréal: CAIS/ACSI, 2007*, 2007.

Zenan Zhai, Dat Quoc Nguyen, and Karin Verspoor. Comparing cnn and lstm character-level embeddings in bilstm-crf models for chemical and disease named entity recognition. *arXiv preprint arXiv:1808.08450*, 2018.

Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 353–362, 2016.

Yongfeng Zhang, Qingyao Ai, Xu Chen, and Pengfei Wang. Learning over knowledge-base embeddings for recommendation. *arXiv preprint arXiv:1803.06540*, 2018.

Yuanzhe Zhang, Xuepeng Wang, Siwei Lai, Shizhu He, Kang Liu, Jun Zhao, and Xueqiang Lv. Ontology matching with word embeddings. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pages 34–45. Springer, 2014.

Huan Zhao, Quanming Yao, Jianda Li, Yangqiu Song, and Dik Lun Lee. Meta-graph based recommendation fusion over heterogeneous information networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 635–644, 2017.

Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. Joint extraction of entities and relations based on a novel tagging scheme. *arXiv preprint arXiv:1706.05075*, 2017.

Peng Zhou, Suncong Zheng, Jiaming Xu, Zhenyu Qi, Hongyun Bao, and Bo Xu. Joint extraction of multiple relations and entities by using a hybrid neural network. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pages 135–146. Springer, 2017.

Leyla Zhuhadar, Olfa Nasraoui, Robert Wyatt, and Elizabeth Romero. Multi-model ontology-based hybrid recommender system in e-learning domain. In *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, volume 3, pages 91–95. IEEE, 2009.