Multimodal Monolingual Comparable Corpus Alignment

Prajol Shrestha

UNIVERSITÉ DE NANTES
FACULTÉ DES SCIENCES ET DES TECHNIQUES

———

ÉCOLE DOCTORALE SCIENCES & TECHNOLOGIES
DE L'INFORMATION ET MATHÉMATIQUES – EDSTIM

Année 2013

Multimodal Monolingual Comparable Corpus Alignment

_____

THÈSE DE DOCTORAT
Discipline : Informatique et applications
Spécialité : Informatique

Présentée
et soutenue publiquement par

Prajol Shrestha

Le 10 octobre 2013, devant le jury ci-dessous

Président	Suresh MANANDHAR, Professeur des Universités, University of York
Rapporteurs	Pascale SÉBILLOT, Professeur des Universités, INSA de Rennes
	Anne VILNAT, Professeur des Universités, Université Paris-Sud
Examinateurs	Suresh MANANDHAR, Professeur des Universités, University of York
Co-encadrante	Christine JACQUIN, Maîtres de Conférences, Université de Nantes

Directeur de thèse :

Béatrice DAILLE, Professeur des Universités, Université de Nantes

Titre Alignement inter-modalités de corpus comparable monolingue

Résumé L'augmentation de la production des documents électroniques disponibles sous forme du texte ou d'audio (journaux, radio, enregistrements audio de télévision, etc.) nécessite le développement d'outils automatisés pour le suivi et la navigation. Il devrait être possible, par exemple, lors de la lecture d'un article d'un journal en ligne, d'accéder à des émissions radio correspondant à la lecture en cours. Cette navigation fine entre les différents médias exige l'alignement des «passages» avec un contenu similaire dans des documents issus de différentes modalités monolingues et comparables. Notre travail se concentre sur ce problème d'alignement de textes courts dans un contexte comparable monolingue et multimodal. Le problème consiste à trouver des similitudes entre le texte court et comment extraire les caractéristiques de ces textes pour nous aider à trouver les similarités pour le processus d'alignement. Nous contributions à ce problème en trois parties. La première partie tente de définir la similitude qui est la base du processus d'alignement. La deuxième partie vise à développer une nouvelle représentation de texte afin de faciliter la création du corpus de référence qui va servir à évaluer les méthodes d'alignement. Enfin, la troisième contribution est d'étudier différentes méthodes d'alignement et l'effet de ses composants sur le processus d'alignement. Ces composants comprennent différentes représentations textuelles, des poids et des mesures de similarité.

Mots-cls Multimodalités, Corpus Comparable, Segmentation Informatif

Title Multimodal Monolingual Comparable Corpus Alignment

Abstract Increased production of information materials like text or audio available (newspapers, radio, audio of television programs, etc..) requires the development of automated tools for tracking and navigation. It should be possible for example, when reading a newspaper article online, to access parts of radio emissions corresponding to the current reading. This fine navigation between different media requires the alignment of "Passages" with similar content within document extracts of different comparable monolingual modalities. Our work focuses on this alignment problem of short texts in a multimodal monolingual comparable context. The problem lies in finding similarities between short text and how to extract the features of these texts to help us find similarities for the alignment process. We contribute to this problem in three parts. The first part tries to define similarity which is the basis of the alignment process. The second part aims at developing a new text representation to facilitate the creation of the gold corpus on which alignment methods will be evaluated. Finally, the third contribution is to study different methods of alignment and the effect of its components on the alignment process. These components include different text representations, weights and similarity measures.

Keywords Multimodality, Comparable Corpus, Information Segmentation

Contents

Introduction

1

Context

Information in this digital world is being created in a rapid pace. From individuals to news corporations, they all are involved in creating digital content in the form of text or audio/video. With all these digital content, various automated tools for systematically accessing information within them is a challenge that has become a necessity. The extensive use of Google's search is a testament of this necessity. Google's search is able to find appropriate files of different media, present in the internet, based on some user given key words. Similar to Google's search, there are many automated systems that could be built to help one track and navigate through various media. In a multimedia context, through these systems, it should be possible for example, when reading a newspaper article online, to access parts of radio emissions corresponding to the current reading. The transcripts of audio recordings in the form of texts, generated by the engines of speech recognition, are necessary to make the bridge with the textual data target. This allows navigation between different documents nature. This fine navigation between different media requires the alignment of Passages, possibly short text segments, with similar content within documents of different modalities. This thesis deals with this alignment of passages or short texts which is part of the continued work on the extraction and categorization of text segments (Barzilay and Elhadad 2003, Hatzivassiloglou et al. 2001, Islam et al. 2008).

Our work focuses on all the aspects of automatic alignment of short texts within documents from different modalities which are related to the same topic and written in the same language. This includes creation of the multimodal monolingual comparable corpus in which alignments will be evaluated, representation of the alignment items, definition of similarity based on which alignments are done and the alignment of the short texts using different similarity measures and clustering methods.

Texts within documents are aligned to each other on the basis of similarity and to facilitate this decision on whether texts are similar or not, there are similarity measures which gives a value on how similar texts are. These measures use internal and/or external resources to compute the value on similarity corresponding to unsupervised and supervised methods respectively. In most of the cases where internal resources are used, the measure of similarity is computed through a matrix which represents

the text in terms of its elements like words. On the other hand, external resources uses WordNet, lexicons etc. in a supervised fashion to extract the semantics of the texts to help decide on their similarity. These external resources are few in number across languages and methods based on them cannot be generalized. In this thesis we focus on the unsupervised method of using internal resources such as the properties of words in terms of distribution, frequency, co-occurrence within the text. All these factors and components related to the process of alignment of multimodal monolingual comparable corpus is studied in this thesis to propose an automatic method for alignment.

Problem Statement

Our work revolves around the objective to propose a method of alignment for multimodal monolingual comparable corpora based on similarity. This alignment field has been under-researched compared to the potential contribution it may offer to the field of information retrieval and extraction. The problem of this task can be divided into three parts as follows:

1 To build a definition of similarity, which will be followed while the alignment process.

2 To create a gold corpus for the evaluation of the alignment methods.

3 To find an alignment method that is able to find similar texts.

Definition of similarity is an objective problem and is a difficult concept to generalise as the definition depends on which application it will be used for. For instance, the definition of similarity between texts to a person looking for paraphrases and a person looking for plagiarized texts would be different. Even though both paraphrases and plagiarized texts are engulfed by the general definition of similar, to have something in common, these texts are different in property which the general definition of similarity is not able to capture. The most general definition of similarity can be thought of as the intuition of something being in common. Focusing on different parts of the text to be common makes the set of similar text segments different in a corpus, for example, reused text, related text, plagiarized text, or paraphrases can all be examples of similar text when the general definition of similarity is changed with what is common. This shows the difficulty and variety in formulating the definition. Our main objective of this alignment process is for information retrieval and extraction hence our definition of similarity will try to help this cause.

The second problem is the creation of the gold corpus with annotations on short text similarity. There are several multimodal corpora like the ones distributed by European Language Resources Association[1] (ELRA) and Linguistic Data Consortium[2] (LDC), but there aren't any gold

[1] http://www.elra.info/Catalogue.html
[2] http://www.ldc.upenn.edu/Catalog/

corpus with annotations on short text similarity to our knowledge. This process of alignment is a difficult and expensive task in terms of human time and efforts. The objective would be not only to create such corpus but also to create it in such a way that the human time and effort required would be drastically reduced.

The third and main problem is the alignment between the segments. Alignment can be taken as a problem of linking segments with their corresponding similarity measure between them or as a clustering problem. We will investigate both types of alignment using different methods and propose a good solution for our objective.

Contributions

This thesis provides a comprehensive overview on the unsupervised alignment process of multimodal monolingual comparable corpus. The process of alignment of these type of corpora are under studied. In this work, we start by studying the foundations of the alignment process. This includes two main parts:

1 Defining similarity, which is the alignment criteria.

2 Creation of the gold corpus.

There are very few works that touch these issues partly because of the complexity of the problem as well as the huge human time and effort required. We also study the performance of various representation of texts on alignment by varying weights and similarity measures on them. In particular, the contribution of this thesis includes the following :

1 Investigating the steps for alignment of short texts and developing a two phase manual method for alignment.

2 Presenting a new text representation method which will further reduce human effort and time for the creation of the gold corpus compared to the manual method.

3 Developing a multimodal monolingual comparable corpus for evaluation.

4 Analysis of similarity in the context of short texts.

5 Investigating the performance of different text representation methods on alignment and the effects of various weights and similarity measures.

Organization

We start the thesis with a general overview on monolingual textual alignment. This includes descriptions of different corpora and the alignment

process. The alignment process includes the identification of text segments to align and also the criteria with which the process of alignment takes place. In chapter 3, we explain how texts are represented using vectors. Different similarity measures are also presented here which uses these text representations to compute the similarity values between texts. The state of the art methods in chapter 4 presents existing research on the segmentation of the corpus in order to receive short texts and alignment methods for short texts including the techniques to evaluate them. With the general overview of our problem and the state of the art methods presented, we explain how the gold corpus is built in Chapter 5. This chapter explains a manual and a hybrid method for the gold alignment of the multimodal monolingual comparable corpus which drastically reduces the human time and effort compared to traditional manual alignment methods. In Chapter 6 we present different methods on automatic alignment of the corpus created in Chapter 5 with their performance evaluated. Finally, Chapter 7 concludes the thesis by discussing the overall contribution of the research in the context of related work in the area. In addition, we present the limitations of the approaches and points to future research directions.

Monolingual textual alignment

2

Contents

In this chapter, we present the concepts that deal with monolingual textual alignments. As monolingual textual alignment is a large field, we focus on the aspects of alignment of multimodal comparable corpora. We define this type of corpora along with its constituting elements and most importantly their alignment. The idea of alignment is presented as a type of arrangement of texts. The prerequisites of alignment and the different types of alignment is explained in detail.

2.1 Corpus

Corpus definition

Text has been the most frequent and used medium for the dissemination of information. A large portion of these texts are present on the web or in machine readable form. A corpus is simply a collection of these texts as stated by Kilgarriff and Grefenstette (2003). This is a very broad definition which is a necessity because nowadays there are many varieties of text collections which are used for Natural Language Processing (NLP) and cannot be confined to some specifics. There are many definitions that are slightly more specific and application oriented (Kilgarriff and Grefenstette 2003) but they fall out of the scope of our NLP work.

There are many organizations that provide linguistically-marked-up corpora but will charge a moderate sum of money. Some of these organizations are listed below :

- Linguistic Data Consortium (LDC)

- European Language Resources Association (ELRA)
- International Computer Archive of Modern English (ICAME)
- Oxford Text Archive (OTA)
- Child Language Data Exchange System (CHILDES)
- Meta Share

Beside these organizations, texts to form a corpus can be collected automatically or manually from the web. The automatic collection of text from the web is called crawling or spidering. This is done by a web crawler (Boleda et al. 2006) which is a program that browse through the web in an orderly fashion to collect the visited pages. In contrast, the manual collection of text are made by manually selecting pages to collect. For example, the Brown Corpus is a hand made collection of abstracts to represent the sample of written American English used in 1961. The Brown Corpus is called a *Sample Corpus* because it contains only a sample of the texts used during 1961. Like the Sample Corpus, there are many different types of corpora depending on the collection of texts each comprise (Pearson 1998), e.g., *General Reference Corpora* which are a large collection of texts which provides comprehensive information about a language like the INaLF corpus for French language; *Specialized Corpora* which are a collection of texts for some special purpose like the KnCr corpus of MEDLINE for clustering scientific abstracts (Pinto and Rosso 2006), *Special Corpora* which are collection of texts that do not contribute to a description of the ordinary language like a corpus of the language of non-native speakers.

Corpus Types

The corpora that are mentioned above are all Monolingual corpora because they all deal with a collection of texts in one language. Corpora that contain collection of texts from several languages are called Multilingual corpora. The popularity of Multilingual corpora has increased as its use in the fields of Machine translation (Lopez 2008) and Lexicon extraction (Fung 1998) has seen progress. Multilingual corpora used for these applications are of mainly two types, i.e., Comparable corpora (Chiao and Zweigenbaum 2002) and Parallel corpora (Fung and Church 1994).

Monolingual and Multilingual Corpus

The Expert Advisory Group on Language Engineering Standards Guidelines [1] (EAGLES) (Sinclair 1996) states the definition of comparable corpora as :

Multilingual Comparable and Parallel Corpora

Definition 2.1.1. A comparable corpus is one which selects similar texts in more than one language or variety.

and parallel corpora as :

Definition 2.1.2. A parallel corpus is a collection of texts, each of which is translated into one or more other languages than the original.

[1] www.ilc.cnr.it/EAGLES96/corpustyp/node1.html

The definition given by EAGLES on comparable corpora is still vague unless we give more information on similarity. EAGLES goes further and gives the nature of similarity by stating:

" *There is as yet no agreement on the nature of the similarity, because there are very few examples of comparable corpora.... The possibilities of a comparable corpus are to compare different languages or varieties in similar circumstances of communication, but avoiding the inevitable distortion introduced by the translations of a parallel corpus.*"

The number and variety of comparable corpora today have drastically increased since the definition of comparable corpus given above. Even with these varieties of comparable corpora, the common feature in them is the property that the texts are somehow similar. This similarity is based on the purpose behind the creation of the corpus. For example, Ji (2009) considers comparable corpora as a collection of documents with similar topics. In the field of machine translation (Munteanu et al. 2004, Hewavitharana and Vogel 2008), comparable corpora are collection of texts that have overlapping information. In the field of contrastive linguistics (Granger 2010), the definition of comparable corpora is given as the collection of texts that have the same criteria such as the time of composition, text category, intended audience, and so on.

Furthermore, some may use the term *parallel corpus* to indicate a combination of both comparable and parallel corpus (Granger 2010), whereas, some may consider parallel corpus to be a special comparable corpus that has the highest level of comparability (Bo Li 2010). Even though there may be differences in the terminology, their lies an underlying difference between the two types of corpora.

Monolingual Comparable and Parallel Corpora

Comparable and parallel corpora in the monolingual aspect is a relatively new concept and have no definitions as in the multilingual scenario. However, the idea of parallel and comparable corpora with respect to the monolingual corpus is similar to that of multimodal corpus. Therefore, the definition of parallel and comparable corpora could be adapted from some definitions for the multimodal corpus. One such definition is presented below which is stated by Granger (2010):

Definition 2.1.3. Parallel corpora are corpora consisting of original texts in one language and their translations into one or more languages.

Definition 2.1.4. Comparable corpora are corpora consisting of original texts in two or more languages, matched by criteria such as the time of composition, text category, intended audience, etc.

These definitions can be modified for monolingual corpora and restated as the following :

Definition 2.1.5. Parallel corpora are corpora that are collection of texts in one language that are different translations of the original texts from one or more languages.

Definition 2.1.6. Comparable corpora are corpora that are collection of original texts in one language, matched by criteria such as the time of composition, text category, intended audience, etc.

These broad definitions reflect how monolingual comparable and parallel corpora are created (Marsi and Krahmer 2007). But, in a monolingual case the distinction between a parallel and a comparable corpus could be difficult. For instance, a corpus that consists of several translated texts inherently contains texts that are matched by the same topic criteria making the distinction difficult. In fact, parallel corpora tends to be a special type of comparable corpora as in the multilingual scenario. Bernardini (2011) has mentioned the difference between comparable and parallel corpus in a monolingual scenario as follows :

"Differences found were interpreted in terms of e.g. a tendency for translated texts to be more/less explicit, unambiguous, repetitive, plain etc., than texts similar along all dimensions, except for the fact that they originated within the target culture instead of being imported from a different one by means of translation"

In such a case, parallel corpora would be the collection of texts which are the exact translations, for instance, there exists two different Dutch translations of parts of the books "*Le Petit Prince*" by Antoine de Saint-Exupéry, "*On the Origin of Species*" by Charles Darwin in the 1^{st} and 6^{th} edition and "*Les Essais*" by Michel de Montaigne which could be made into a parallel corpus (Marsi and Krahmer 2007). These parallel corpora, being a translation, contain texts that are more or less the exact same information and are more likely to be repetitive and unambiguous. Furthermore, comparable corpora would be the collection of similar texts in the same language which may contain different information with less repetition, for example, in the case of encyclopedia where there are some articles about the same topic but written for different groups of audiences, for instance, one is targeted for adults, called the Encyclopedia Britannica, while a simpler version targets younger audiences, called Britannica Elementary (Barzilay and Elhadad 2003), another good example is the collection of news articles on the same topic, for instance, articles from different news agencies about the earthquake at Fukushima .

Multimodality

The corpora that we have been discussing are all collection of written texts which use visual modality for communication. Humans are able to communicate through other production and sensory modalities, i.e. voice, auditory, tactile, olfactory, and gustatory. The modalities that are extensively used for communication and which are well documented are visual, voice, auditory and their combination. Visual modality provides communication through written text, images as well as gestures while auditory and voice modality provides communication through the medium of speech. Videos on the other hand use the combination of visual, voice and auditory modalities for communication. Multimodal corpus is a collection of texts whose source is the combination of more than one modality. Recently there are many studies as well as workshops

Multimodal Corpora

on Multimodal corpora[2]. There are no formal definitions of Multimodal corpora but a broad definition is given by Allwood (2008) and says :

Definition 2.1.7. A first attempt at a definition might now be to say that a multimodal digitized corpus is a computer-based collection of language and communication-related material drawing on more than one sensory modality or on more than one production modality.

This section puts forward the idea about what is a corpus and its basic types. The different varieties of corpora that are presented here give rise to other types of corpora with the different combinations of these basic types. For instance, combining the property of multimodality, monolinguality, comparability will produce a corpora which is called Multimodal Monolingual Comparable Corpus. This corpus may contain multimodal textual data of written text and transcribed text of audio or videos from a single language which are related to the same topic. This combination shows that the different corpora that are mentioned in the literature are basically the properties of the corpora which can be combined together to create a new type of corpus for some specific task.

Multimodal Monolingual Comparable Corpus

2.2 Alignment

In NLP, alignment is the organization of objects in the form of mapping, linking or grouping. The concept of alignment is used in many NLP tasks, for instance, in Statistical Machine Translation (SMT) word, phrases, or sentences are linked from one language to another; Text to Speech (TTS) where texts are mapped to its phonetic transcriptions; Automatic Speech Recognition (ASR) where speech signals are linked to phonemes, words, phrases, or sentences; Summarization and information extraction where similar texts are grouped together. For different tasks, the object and the manner of their organization is different. Among the range of objects, we focus on the alignment of texts.

Defining Alignment

In the field of NLP, when textual alignment is mentioned, it mostly refers to alignments between parallel texts where texts from one language are mapped to its translation or translations. One of the main reasons behind this is the vast literature available on SMT which uses aligned parallel corpora (Lopez 2008). Parallel texts were first tried in the fifties, but due to the limitations on computers and the unavailability of texts in digital form, the results were not encouraging (Véronis and Langlais 2000). In the eighties, with the increase in computation and storage power as well as the availability of digital texts and advancements in statistical methods, interest in text alignment emerged. Towards the end of the 1980s, example-based or memory-based approach of machine translation was used, where, extraction and selection of equivalent words, phrases or word groups were performed on aligned parallel texts. These parallel texts were either aligned by statistical methods or by rule-based methods (Hutchins 2007). Since then, textual alignments on different levels,

[2]www.multimodal-corpora.org

Bilingual / Multilingual Alignment

such as, words, phrases, group of words, sentences, group of sentences, has been in a rise to aid different applications in the field of NLP. These alignments between two or more languages are called bilingual or multilingual alignments respectively. One of the first automatic alignments dealt with aligning sentences (Kay 1991) and was devised to align sentences between two languages to aid MT.

Monolingual Alignment

Similar to bilingual alignment, textual alignment between the texts in the same language is known as Monolingual alignment. Various types of applications, such as *text summarization, information retrieval/extraction, paraphrase detection, topic wise navigation through text* and *text generation*, use monolingual alignment at different textual levels. These alignments are a vital part for the development and testing of these applications. The alignment of texts at different level depends on the application: summarization might require alignments between short text segments while paraphrase detection/extraction might require alignments between sentences or phrases.

Various Monolingual Aligned Corpora

The field of NLP requires a wide range of aligned corpora with different types of alignments to test and validate the automatic methods but there are only a few publicly available corpora with monolingual alignments. There are some standard aligned corpora available, created manually and or automatically by collecting existing texts written by humans or artificially by generating machine made text, which are listed below :

- TDT corpus[3], for Topic detection and tracking applications for English language which is manually created.
- PAN-PC-09 (Barrón-Cedeño et al. 2010), for plagiarism detection, is artificially created due to the fact that natural text on plagiarism is hard to collect. It consists 90% of monolingual English and 10% of cross-language mostly German and Spanish plagiarisms.
- METER corpus (Gaizauskas et al. 2001), for the detection of text reuse in British English, manually created.
- MSRPC corpus (Dolan et al. 2004), for the detection of paraphrase in English language, automatically created.

These corpora are crucial for the development of methods and algorithms that will be able to automatically align texts for specific tasks. But these aligned corpora are scarce and most of the ones that are available are not free. On top of this, new tasks are constantly being built which requires new aligned corpora. These corpora will have to be created and aligned to develop and validate solutions for each specific tasks.

Types of Alignment

The task of alignment or the linkage of texts can be done in two ways. The first method connects pair of text units based on some criteria and are known as pairwise alignment. For example, in the task of information retrieval, pairwise alignment is done to retrieve text pairs that have the

[3]http://projects.ldc.upenn.edu/TDT-Pilot/

same information content. The second method creates different groups of text units such that the texts in each group are connected to each other on the basis of some criteria. This method is called group-wise alignment. These two types of alignments are shown in the Figure 2.1

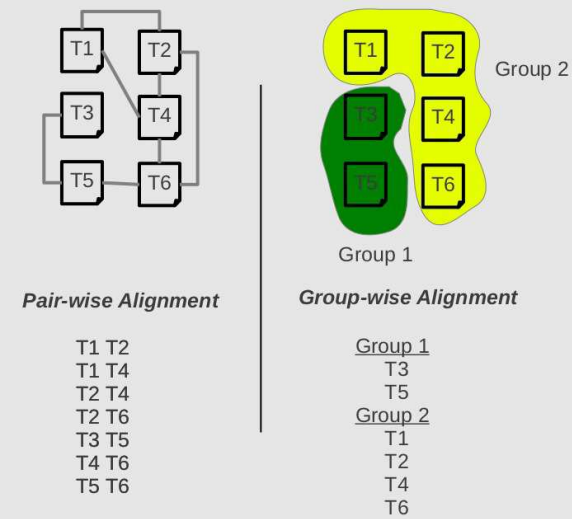


Figure 2.1 – *The pair-wise and group-wise alignments are shown on the left and right side respectively. In the pair-wise alignment links are shown by joining pair of texts that satisfy the agreed criteria and the pairs are listed below. The group-wise alignment shows the grouping of texts that satisfy the agreed criteria and the list consists of text present in each group.*

These two alignments are done manually or automatically. In most of the cases, corpora are manually annotated while there are some which are automatically annotated. Even though automatic alignment require no human effort, the alignment accuracy is lower than manual alignment which requires some human checking in order to create a gold standard. This is the main reason why manual alignment is preferred to an automatic alignment to create a standard aligned corpora. The group-wise alignment is the easiest in terms of human effort. This is because the alignment process requires the annotator to read each text only once. Each time the annotator reads a text, the annotator assigns it to a group depending on the alignment criteria and the set of groups (Pinto and Rosso 2006). In most of the group-wise alignment task, the number of groups are known beforehand, but groups can also be dynamically incremented as the alignment process is carried out but it requires to read the texts again. In contrast, the pair-wise alignment process requires each text to be compared with every other text present in the collection to make alignments. Thus, the total number of comparisons required to align a set of n texts in the worst case is :

$$\frac{n(n-1)}{2} \tag{2.1}$$

Even a small set of 100 texts, would result into 4,950 comparisons, and tends to increase quadratically as the number of text increases. In practice, there are hundreds or thousands of texts to be aligned. The manual pro-

cess of the pairwise aligning of these texts would consume a huge amount of human effort and time. There are also situations where the pair-wise comparisons would not follow the amount given by equation 2.1. These situation arise while performing pairwise alignment between parallel texts where we know the text units have mostly a one to one relationship. In such a case, the occurrence of the two text units is approximately relatively known which reduces the number or comparisons close to the numbers generated by the group based alignment.

Manual annotations are expensive due to the large human effort and time that is required. That has lead to alternative solutions such as crowd sourcing. Crowd sourcing is a process that involves outsourcing tasks to a distributed group of people. Amazon Mechanical Turk[4] is one of the mostly used tool created for crowd sourcing and has been recently used for the manual annotation process (Wang et al. 2011). Even though crowd sourcing is an option to reduce the cost, the quality of the alignments is difficult to assure because of the problem of verifying that each participant has been following the alignment criteria correctly.

There have been research on automatic methods for annotation, but they have not reached a performance for a practical use. There are available corpora that have been automatically aligned, for example, the MSRPC corpus which contains paraphrases selected using a classifier with limited features; the PAN-PC-09 corpus which contains alignments between plagiarised texts using certain modifications to the texts creating artificial plagiarised texts. However, the aligned text units in these automatically aligned corpora are generally only a subset of the actual problem as they don't contain alignments that could not be captured by the classifiers (Dolan et al. 2004).

Prerequisites for Alignment

Whether the alignment is carried out automatically or manually, there are two prerequisites for any alignment process. These prerequisites are listed below:

1 Identification of the text units that has to be aligned, for instance, the text unit could be a word, term, sentence, text segment or even a document.

2 To define the criteria of alignment, i.e, the basis on which aligned text units will be chosen, for instance, text units could be aligned on the basis of plagiarism, or having the same topic.

These two prerequisites are defined depending on the task in hand. For example, in the task of paraphrase detection, the text unit could be phrases or sentences and the criteria is that the text units should be paraphrases of each other. Once these two entities are defined the process of alignment can begin.

In the following sections, we explain how text units are generated and

[4]www.mturk.com

how the alignment criteria may be defined depending on the objective.

2.2.1 Text units

Identifying the required text units, or in other words the text unit boundaries, for alignment is one of the task that is necessary before the process of alignment. This is a subjective matter and depends on the objective of alignment. Text unit boundaries are the start and end of the text unit. One of the most well known text unit boundaries are the boundaries of linguistic units. There are a range of orthographic conventions used in written languages to denote the boundaries between linguistic units such as word, sentence, and text segment boundaries. For example, Amharic has explicit markers indicating the word and sentence boundaries, whereas Thai has no boundaries with few cues to indicate different segments at any level. Most of the written languages lie between these two languages that have text unit markers indicating different text levels. English, at word boundaries use whitespaces between most words and punctuation marks at sentence boundaries. But they are not always sufficient to segment the text as these markers are known to be ambiguous (Dale 2010).

Segmentation of Text

Segmentation is a process or method to find text units, the alignment units, by finding their boundaries. Word and sentence segmentation in English is done using segmentation tools.OpenNLP library[5] (Koeva et al. 2012) consists of a sentence detector and tokenizers which help segment sentences and words using rule-based and/or machine learning techniques [6]. Beside words and sentences, other text units may include document segments, discourse segments as well as short text segments which include groups of sentences. Having documents as a text unit, in most cases, would be the best case scenario for segmentation where the text units are physically separated as they are found in different files requiring no segmentation. Segmentation of text units such as discourse segments and short text segments that are part of the document containing a group of sentences may be a problem. In such a case, the text segments might not be easily identified which makes the task of segmentation by identifying their boundaries difficult. Despite this difficulty, some studies that require segmentation have shown that paragraph marker are one of the best clues for text segments (Hatzivassiloglou et al. 1999).

There are three ways in which segmentation can be performed to extract text segments. The first method of segmentation is the manual segmentation in which the texts are manually analysed to find segment boundaries. For example, in this method, a sentence or a group of sentences that convey a particular information or sentences that are coherent and related to the same subtopic are marked to indicate a text segment. This process is similar to the manual text alignment in terms of human effort. Even though it creates accurate segmentations, it is difficult and time consuming which will make the overall task of alignment even more

[5]OpenNLP library can be found at: opennlp.apache.org

[6]Documentation of OpenNlP : http://opennlp.apache.org/documentation/manual/opennlp.html

daunting.

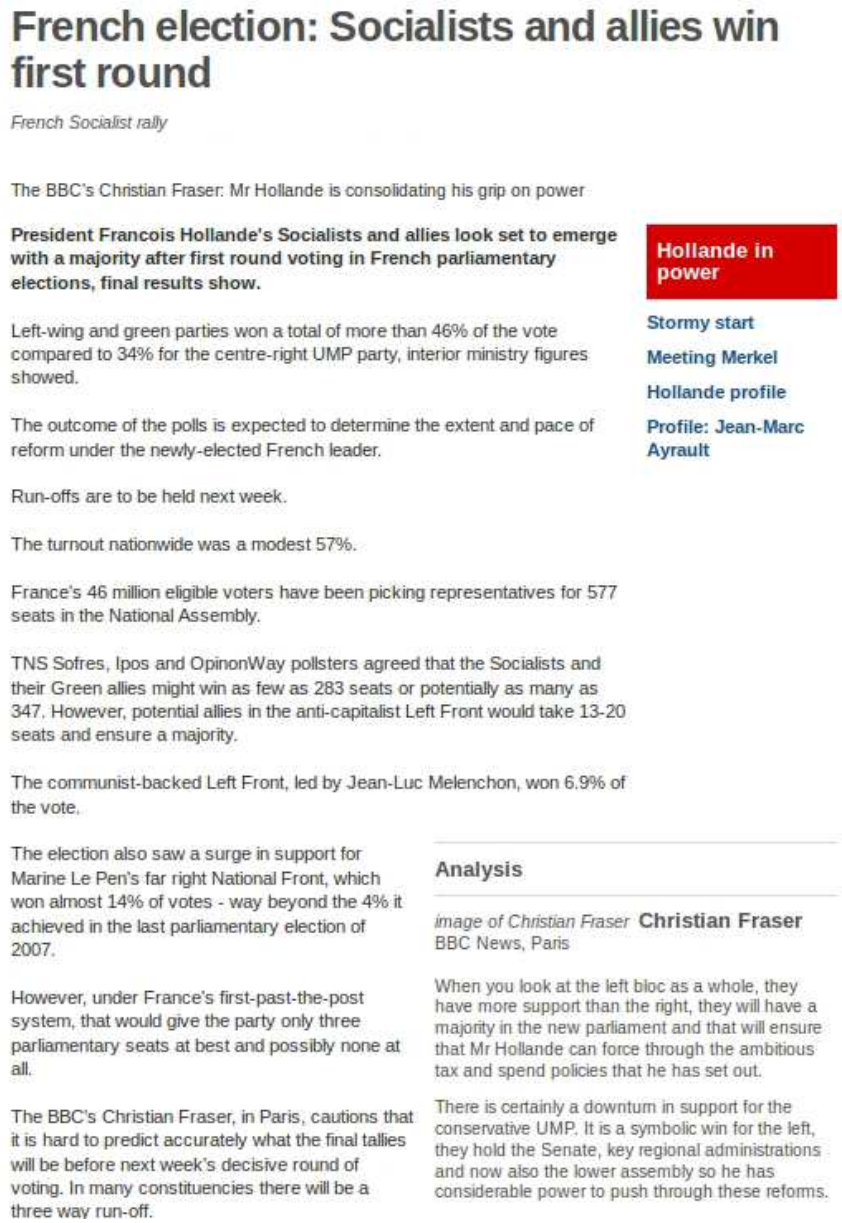
Segmentation Methods for Text Segments

The second method of segmentation is the naive method which considers each natural segmentation in the structure of the text, i.e. paragraph marking, as a text segment boundary (Hatzivassiloglou et al. 1999). These natural boundaries are created to make distinct partitions of the text based on various reasons. Most of the time, structural partitions are used to control the flow of information which results in the control of ideas that are being presented. Depending on the text, it is possible to take the natural structural partitions of the text as text segments where mostly a certain idea or information is present. These types of segments are mainly present in electronic news articles where small paragraphs with few short sentences are created to convey an idea or information in a precise manner. An example of such type of text is shown in Figure 2.2[7] which is a snippet from a BBC news article that shows text segments, separated by a new line indicating the paragraph markings.

The boundaries depends on the modality and the genre of the documents. For instance, in transcribed texts, there are two distinct natural boundaries that could be easily found and considered boundaries of text segments. These are the boundary of short texts and boundary of turn of speakers. The nature of short texts in the transcripts are similar to paragraphs in written texts where they are created to control the flow of information from a single speaker which consists of few sentences. The turn boundary on the other hand indicates the information presented by each speaker. These boundaries are necessary and present when two or more speakers are involved. The combination of these two boundaries can be seen in Figure 2.3[8]. In this figure, at the beginning the host of the show introduces the guest of the show and has paragraph like segmentation. However, towards the end of the figure the host starts the conversation with the guest which creates segments based on speaker turns.

The third method of segmentation is using automatic methods. These automatic methods are used when the choice of text segments cannot be determined by the natural boundaries of sentence or paragraphs. Figure 2.4[9] shows an example of a text that contains large paragraphs which may not be a suitable text segment for certain tasks like text generation and would require segmentation. The automatic methods, mainly use two properties of texts for segmentation. These properties are lexical cohesion and discourse cues. Lexical cohesion is the lexical relationship that exist within texts that makes the text coherent and gives its meaning (Halliday and Hasan 1976a). Lexical cohesion includes properties of word reiteration, co-occurrence, and statistics present within texts. Using these properties, the automatic segmentation methods determine which sentences are coherent to present it as a text segment (Kaufmann 2000).

[7] http://www.bbc.co.uk/news/world-europe-18388273
[8] http://transcripts.cnn.com/TRANSCRIPTS/1209/21/ampr.01.html
[9] http://www.imf.org/external/np/prsp/prsp.aspx



French election: Socialists and allies win first round

French Socialist rally

The BBC's Christian Fraser: Mr Hollande is consolidating his grip on power

President Francois Hollande's Socialists and allies look set to emerge with a majority after first round voting in French parliamentary elections, final results show.

Left-wing and green parties won a total of more than 46% of the vote compared to 34% for the centre-right UMP party, interior ministry figures showed.

The outcome of the polls is expected to determine the extent and pace of reform under the newly-elected French leader.

Run-offs are to be held next week.

The turnout nationwide was a modest 57%.

France's 46 million eligible voters have been picking representatives for 577 seats in the National Assembly.

TNS Sofres, Ipos and OpinonWay pollsters agreed that the Socialists and their Green allies might win as few as 283 seats or potentially as many as 347. However, potential allies in the anti-capitalist Left Front would take 13-20 seats and ensure a majority.

The communist-backed Left Front, led by Jean-Luc Melenchon, won 6.9% of the vote.

The election also saw a surge in support for Marine Le Pen's far right National Front, which won almost 14% of votes - way beyond the 4% it achieved in the last parliamentary election of 2007.

However, under France's first-past-the-post system, that would give the party only three parliamentary seats at best and possibly none at all.

The BBC's Christian Fraser, in Paris, cautions that it is hard to predict accurately what the final tallies will be before next week's decisive round of voting. In many constituencies there will be a three way run-off.

Hollande in power

Stormy start

Meeting Merkel

Hollande profile

Profile: Jean-Marc Ayrault

Analysis

image of Christian Fraser **Christian Fraser**
BBC News, Paris

When you look at the left bloc as a whole, they have more support than the right, they will have a majority in the new parliament and that will ensure that Mr Hollande can force through the ambitious tax and spend policies that he has set out.

There is certainly a downturn in support for the conservative UMP. It is a symbolic win for the left, they hold the Senate, key regional administrations and now also the lower assembly so he has considerable power to push through these reforms.

Figure 2.2 – *The natural structural text segments, i.e. paragraphs, with few sentences present in the BBC news article "French election: Socialists and allies win first round"*

CNN'S AMANPOUR

Interview with Aung San Suu Kyi; Interview with Bernard Henri-Levy

Aired September 21, 2012 - 15:00:00 ET

THIS IS A RUSH TRANSCRIPT. THIS COPY MAY NOT BE IN ITS FINAL FORM AND MAY BE UPDATED.

CHRISTIANE AMANPOUR, CNN HOST: Hello again, everyone, I'm Christiane Amanpour, and welcome to the weekend edition of our program. We begin with Aung San Suu Kyi, one of the world's most revered advocates for democracy. This year, she was finally freed and able to collect the Nobel Peace Prize that she'd won back in 1991.

She has spent almost 20 years under house arrest in Myanmar, also known as Burma, isolated from the world and from her family by the brutal and oppressive military regime there. During her captivity, Suu Kyi lost her husband to cancer and she was estranged from her two young sons, who were forbidden to visit her.

But in the last year, Suu Kyi's long struggle has finally paid off. The country's new president, Thein Sein, freed her from detention and instituted a series of economic and political reforms. She won a seat in parliament and now Suu Kyi, the icon, has become Suu Kyi, the politician. She's stepped off the pedestal and into the fray.

On Wednesday, Aung San Suu Kyi was in Washington, meeting with President Obama and his dog, Bo, and as you look at this picture, bear in mind that for the whole time she was under house arrest, Aung San Suu Kyi refused to get a dog for company because, she said, it wouldn't be fair for a dog not to run free.

She received a hero's welcome at the U.S. Congress, where she accepted a Congressional Gold Medal. The ceremony was also broadcast at home in Myanmar, the first time that the state broadcaster had aired footage of Suu Kyi overseas. Aung San Suu Kyi has just returned to New York, where she lived and worked briefly 40 years ago, and I caught up with her in a break from her whirlwind tour.

(BEGIN VIDEO CLIP)

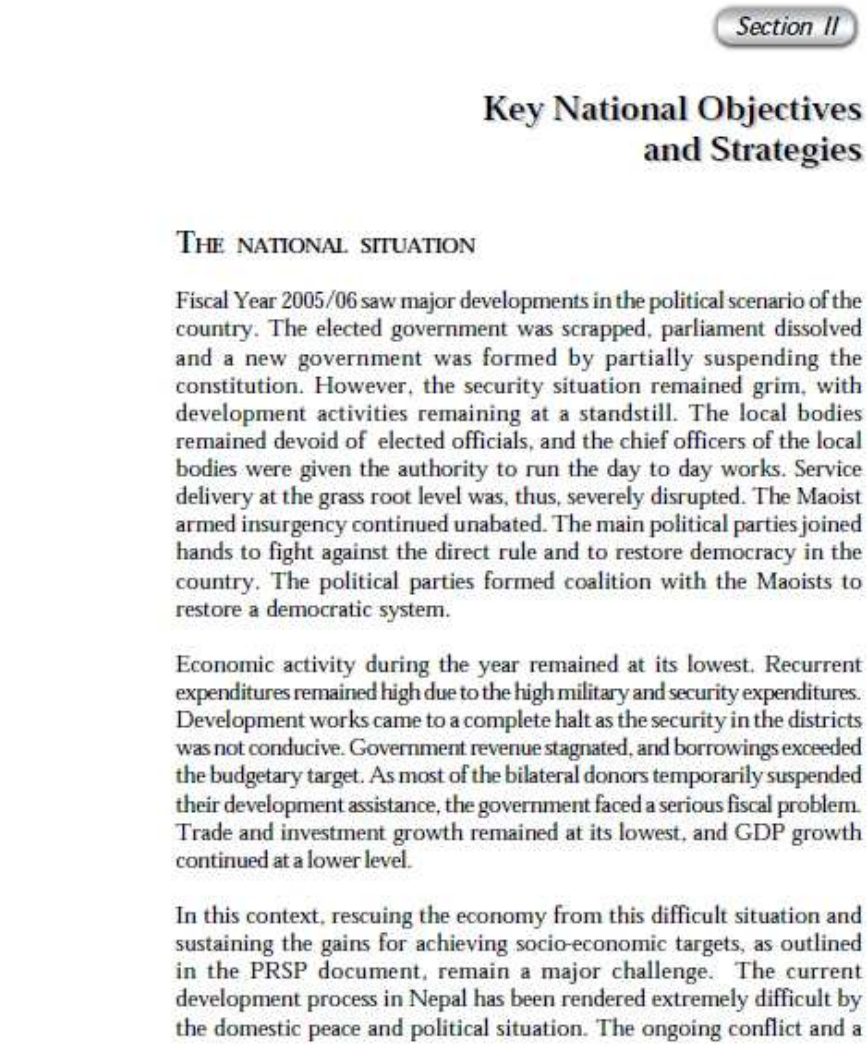
AMANPOUR: Aung San Suu Kyi, thank you for joining me.

AUNG SAN SUU KYI, BURMESE ACTIVIST: It's a pleasure.

AMANPOUR: I see you in these amazing public events now, accepting finally the Congressional Gold Medal, the Nobel Peace Prize. You get a hero's welcome. You looked visibly pained when people are standing up in these prolonged standing Os. Is it weird for you?

SUU KYI: No, it's -- I appreciate it very much. But sometimes I feel a little embarrassed.

Figure 2.3 – *The natural structural text segments, i.e. paragraphs and turn of speakers, with few sentences present in the transcript.*



Section II

Key National Objectives and Strategies

THE NATIONAL SITUATION

Fiscal Year 2005/06 saw major developments in the political scenario of the country. The elected government was scrapped, parliament dissolved and a new government was formed by partially suspending the constitution. However, the security situation remained grim, with development activities remaining at a standstill. The local bodies remained devoid of elected officials, and the chief officers of the local bodies were given the authority to run the day to day works. Service delivery at the grass root level was, thus, severely disrupted. The Maoist armed insurgency continued unabated. The main political parties joined hands to fight against the direct rule and to restore democracy in the country. The political parties formed coalition with the Maoists to restore a democratic system.

Economic activity during the year remained at its lowest. Recurrent expenditures remained high due to the high military and security expenditures. Development works came to a complete halt as the security in the districts was not conducive. Government revenue stagnated, and borrowings exceeded the budgetary target. As most of the bilateral donors temporarily suspended their development assistance, the government faced a serious fiscal problem. Trade and investment growth remained at its lowest, and GDP growth continued at a lower level.

In this context, rescuing the economy from this difficult situation and sustaining the gains for achieving socio-economic targets, as outlined in the PRSP document, remain a major challenge. The current development process in Nepal has been rendered extremely difficult by the domestic peace and political situation. The ongoing conflict and a

Figure 2.4 – *The natural structure of large text segments in the report of IMF.*

On the other hand, discourse cues deals with the structure of the discourse. Human communication is characterized by distinct discourse structure which is used for various reasons including managing interaction between participants, mitigate limited attention, and indicating shift in topics (Grosz and Sidner 1986). These cues are mostly evident and can be exploited in genre of journalistic or technical text and programs. Beside these explicit cues, linguistic cues are also present within the discourse that deals with linguistic properties such as occurrence of anaphoric expressions, conjunctives, topical markers and type of sentences to express structural and semantic relationships between text to help find boundaries for text segments (Mochizuki et al. 1998). In Chapter 4, we present state of the art automatic methods that use these lexical and discourse properties for segmenting texts.

2.2.2 Alignment Criteria

As mentioned in section 2.2, there are many applications of text alignment. These applications determine the size of the text units and the criteria of alignment, i.e. the basis of alignment. The main applications are information retrieval, text categorization, paraphrase detection, summarization and language simplification. Each of these applications have its own alignment criteria. In information retrieval, the criteria is that the text units should be similar to each other; in paraphrase detection, the text units should be a paraphrase of each other; in text categorization, the text units should be from the same domain or theme; in the case of text summarization, the text units should have the same topic; and in language simplification, the text units should be the simpler form in terms of vocabulary and sentence structure.

Text Similarity

Our task is focused towards information retrieval, where we try to find similar text segments as mentioned in chapter 1. Similarity is a difficult concept to define and is often used as a general term that covers a wide range of phenomena. Even with this difficulty, there are many similarity measures (Barron-Cedeno et al. 2009) like the cosine similarity, used to measure similarity between texts, but these measures do not define similarity, they rather assign a value of similarity based on overlap of terms.

In psychology, similarity is well formalized and captured in formal models such as the set-theoretic model (Tversky 1977) or the geometric model (Widdows 2004). In an attempt to overcome the traditional loose definition of text similarity, Bär et al. (2011) uses the concept framework based on conceptual spaces (Gärdenfors 2000). In this method, texts are represented in three geometric spaces of *Structure* like the order of sections, *Style* as in grammar, usage, lexical components, and *Content* addressing all facts and their relationships with a text. Table 2.1 gives an overview of common NLP tasks and their relevant dimensions using which human make their judgement on the decision of similarity.

Another point of view on similarity may arise with the concept of

Task	structure	style	content
Authorship Classification		✓	
Automatic Essay Scoring	✓	✓	✓
Information Retrieval	✓	✓	✓
Paraphrase Recognition			✓
Plagiarism Detection		✓	✓
Question Answering			✓
Short Answer Grading	✓	✓	✓
Summarization	✓		✓
Text Categorization			✓
Text Segmentation	✓		✓
Text Simplification	✓		✓
Word Sense Alignment			✓

Table 2.1 – *Classification of common NLP tasks with respect to the dimensions of text similarity : structure, style, and content*

meaning of texts. This could be made on the basis of the definition of meaning. Meaning itself is difficult to extract from text and theories such as meaning-text theories try to find the components and explain how meaning is expressed in languages (Milicevic 2006, Melčuk 1981). The meaning-text theory proposes a structure called the Meaning-Text Model (MTM) on how meaning is transformed into text. The structure consists of seven levels starting from the meaning level or the semantics to the speech level or the deep phonology as shown in Figure 2.5.

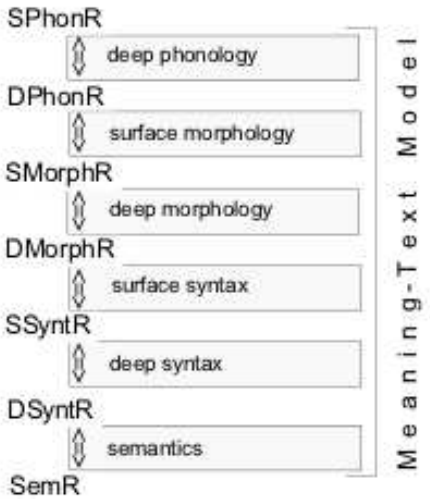


Figure 2.5 – *Structure of a MTM*

The MTM shows the different steps a meaning takes to be refined into speech and vice versa. This complex system would be difficult to use to analyse meaning from text but there are efforts throughout the linguistic community to work on individual levels to understand the working of such phenomenon.

The concept of similarity is subjective. A sentence, for instance, may contains more than one information and the similarity can be based on any of these information, or based on their combination which makes similarity subjective. Goodman (1991) gives a good example, regarding the baggage check at an airport, on the issue of subjectivity of the definition of similarity : While a spectator might compare bags by shape, size, or color, the pilot only focuses on a bag's weight, and the passenger compares them by destination and ownership. Similarly, text also have certain inherent properties that need to be considered in any attempt to judge their similarity which makes defining similarity difficult. However, Lin (1998a) has proposed a general and universal definition of similarity that could be applied to all applications and can be applicable as long as there is a probabilistic model. The intuition behind this definition and two other definitions that have been used in different alignment tasks are listed below.

Definition 2.2.1. Two sentences are similar if they contain at least one clause that expresses the same information. (Barzilay and Elhadad 2003)

Definition 2.2.2. Two paragraphs are similar if they contain "common information". This was defined to be the case if the paragraphs referred to the same object and the object either (a) performed the same action in both paragraphs, or (b) was described in the same way in both paragraphs. (Hatzivassiloglou et al. 2001; 1999)

Definition 2.2.3. Two texts are similar on the basis of these intuitions:(Lin 1998a)

- **Intuition 1:** The similarity between A and B is related to their commonality. The more commonality they share, the more similar they are.
- **Intuition 2:** The similarity between A and B is related to the differences between them. The more differences they have, the less similar they are.
- **Intuition 3:** The maximum similarity between A and B is reached when A and B are identical, no matter how much commonality they share.

All the definitions presented above focus on what is common between the text segments to call them similar. This focus on what is common is also the difference between them. Definition 2.2.1 and 2.2.2 states what should be common where as definition 2.2.3 gives no information about it and therefore, is the most general definition among them. Definition 2.2.1 and 2.2.2 are not general enough and could be difficult to apply to all types of text segments. Definition 2.2.1 is specific to sentences and paragraphs with more than one sentence cannot be considered similar on the basis of the same information within clauses. Definition 2 considers similarity on the basis of objects and there would be paragraphs for which the information about the objects alone do not represent the meaning of the paragraph as in the following paragraph:

French television said Diana was being pursued by paparazzi when the crash occurred, and French Interior Minister Jean-Pierre Chevenement said police were questioning seven photographers as part of a criminal investigation into the accident.
In this paragraph, the object, paparazzi, does not perform any action nor does the description of the paparazzi as photographers represents the paragraph.

Intuition 1 of definition 2.2.3 is a better choice for the definition of text similarity once the term commonality is defined because of its general and intuitive nature. In most of the NLP applications, for instance, summarization, paraphrase detection, and information extraction, similarity is based on the common information between texts. We define the term 'commonality' in the definition on the basis of information and can be stated as follows :

Similarity Definition

Two text segments are similar if at least one of the "main information" that the paragraph conveys is common.

Intuition 2 considers the differences between texts to be an element of how similar they are but we only focus on commonalities to determine how similar they are. Therefore, differences also depend on commonality for instance, the fewer the commonality present between texts the more differences they have. Intuition 3 is partially correct as identical text segments are definitely similar to it's maximum as they will share the same "main information".

A text segment may have more than one "main information" and for us a minimal of one commonality in the main information would make the text segment similar. Here is an example where we compare two paragraphs for similarity:

Paragraph I (PI):

William and Harry, with their father Prince Charles and their grandmother Queen Elizabeth, are thought likely to remain in seclusion at Balmoral Castle in Scotland until Saturday's ceremony.

Paragraph II (PII):

The royal family remained at Balmoral in Scotland Tuesday, with reports that Charles and his younger son Prince Harry went for a walk in the afternoon. It was not clear when they would return to London.

In the paragraph pair given above, the common information present is the information about the royal family secluded at Balmoral Castle. These paragraphs also indicate the importance of the context while finding the concept. Here proper names like "William" and "Harry" with the help of the phrases like "father Prince Charles" and " Charles and his younger

son" help understand the concept that the entities represent.

Here is another example of finding the similarity between paragraphs which is less intuitive at the first glance: Paragraphs to compare are :

Paragraph III (PIII):

Dodi Al Fayed's father, Harrods Department Store owner Mohammed Al Fayed, arrived here immediately after learning of his son's death.

Paragraph IV (PIV):

Bernard Dartevelle, a lawyer for Mohamed Al Fayed, Dodi Fayed's wealthy businessman father and also the owner of the Hotel Ritz, said the revelation "changes absolutely nothing." He spoke of an "ambience of harassment" created around Diana and Fayed by the constant presence of paparazzi.

In these two paragraphs, there are some information in common but the main information is not. Because of this they are not considered similar. In paragraphs PIII and PIV, the information that Dodi Al Fayed's father is a businessman is common but this information is not the main information because these information are present to support the main information given by the paragraph and not the main information itself. Even though we consider this information not to be the main information, this decision is subjective and depends on the reader or the application. For instance, if this information is new to a reader, then the reader could consider this as a main information making the texts similar. In another scenario, the reader may consider it to be a main information, but comparing it to the other main information present in the text, the reader could consider this information to be less significant making the text not similar. These subjective point of views of the readers have sometimes made the alignment criteria to be further categorized to make the alignments multivalued. Categorization can be done using the degree of fulfilment of the alignment criteria, for example, two texts could be ranked between 0 to 5 to show how strongly the criteria of alignment is fulfilled. Another example would be to categorise the texts as exactly similar, moderately similar, and not similar.

Despite the multivalued alignments, similarity in most cases tend to have a binary relationship indicating at least one main information is common or none and texts that are similar must be bidirectional but do not necessarily satisfy transitivity. These properties are shown below where A, B and C are text segments:

$$\text{If } A \sim B \text{ then } B \sim A \text{ and,}$$

$$\text{If } A \sim B \text{ and } B \sim C \text{ then not necessary that } A \sim C$$

The transitivity property of similarity may not always work while dealing with text segments because of the fact that some text segments may contain more than one main information creating overlaps of text segments based on various main information. This is also the reason why

the binary measure can be transformed into a continuous measure as the number of main information within a text segment can be counted.

Conclusions

In this chapter, we gave an overall view of the alignment of texts and how alignment is performed. Alignment of text is the organization or state among text segments with a predefined criteria. There are three entities that take part in aligning texts which are the unit of text to be aligned, the defined criteria which will be the basis of alignment, and the algorithm or method to align. In this thesis, we will align text segments, which will be small groups of sentences containing certain information, from different modalities (newspaper, transcribed broadcast news) of the same language. These text segments will be aligned on the basis of similarity. Our definition of similarity is intuitive and indicates that two segments are made similar if they have at least one common main information. These two entities are the building blocks of the alignment process. Once these are made the alignment process can be started.

Text segments can be aligned in two ways, one by aligning texts in pairs and the other in groups. These alignments can be performed either manually or automatically. Manual alignment is more accurate than automatic alignment which is why they are used to gold standard corpora. But manual alignment is impractical as it is difficult in terms of human effort and time which motivates the automatic alignment of texts.

In the next chapter, we explain different methods of representing texts as vectors which tries to extract the semantics of the texts. We also present the state of the art similarity measures that use these text representations to give a similarity value which are vital parts of the automatic alignment process.

Text Representation and Automatic Alignment

3

"Language fails not because thought fails, but because no verbal symbols can do justice to the fullness and richness of thought. If we are to continue talking about "data" in any other sense than as reflective distinctions, the original datum is always such a qualitative whole."

-John Dewey, *The Symposium (1930)*

Contents

This chapter presents four different methods to create mathematical representations of texts. These representations of texts are vectors which are created using the terms of the texts in order to capture its semantics. They allow the identification of texts that are semantically close or in other words similar to each other. This closeness can be computed using various similarity measures described in this chapter. Using these mathematical representations of text and similarity measures, texts that are semantically close are able to be automatically aligned. This process of automatic alignment is also presented in this chapter.

3.1 Text Representation

Texts can be mathematically represented using vectors called text vectors. These vectors are created using the terms present in the texts. A text vector corresponding to a text can be considered to be the description of its content. Each dimension of these vectors corresponds to the dimensions of some semantic space where the projection of texts can convey the relative similarity relations between them. Here, we present four different methods to create text vectors to represent a set of texts in semantic spaces. These methods are, Vector Space Model (VSM) (Salton 1979), Latent Semantic Analysis (LSA) (Landauer and Dumais 1997), Principle Component Analysis (PCA) (Jolliffe 2002) and Independent Component Analysis (ICA) (Honkela et al. 2010). VSM and LSA have originated to deal with NLP tasks whereas PCA and ICA are widely adopted in the field of pattern recognition and signal processing before their use within NLP tasks. Among the four methods, VSM is the oldest method used for text representation and the other methods can be considered as extensions. Before going into detail about each of these methods we first have to define the terms of the text and their weights which are the building blocks of the text representation.

3.1.1 Term selection and their weights

Terms

The configuration of the semantic space, that determines the representation of texts, depends on the terms present in the texts. Terms are entities within the text that together gives meaning to it. The simplest form of terms are words but there are other choices such as phrases, features, n-grams and so on. The choice of terms is crucial to obtain an accurate representation. This dependence leads towards efforts in the selection process of terms and a scheme to weight them. These weights of terms indicate how important each term is to represent the text. The idea to select terms and weighting them is to select the most discriminative set of terms and weights which creates the vectors that are suitable for the identification of similar texts. Term selection also helps in reducing the dimension of the semantic space which in turn helps in dealing with large collection of texts. There is no one method that is used for term selection process as it is a tricky problem, but there have been attempts to select terms for the better identification of similar texts and to reduce the dimensionality of vectors while dealing with large corpora.

Term Selection

Selecting Terms

For the identification of similar texts, simple term selection techniques such as selecting high frequency terms in a document, term frequency (TF), selecting terms that appear in a high number of documents, document frequency (DF), and Transition Point (TP) (Pinto et al. 2007) which is based on Zipf Law (Zipf 1949) are used. Term selection for the purpose of dimensionality reduction, especially in the task of text classification, is based on information theoretic functions such as DIA association factor (Fuhr and Buckley 1991), chi-square (Caropreso et al. 2001, Schütze

Function	Mathematical Form
DIA association factor	$P(c_t\|t_k)$
Information gain	$\sum_{c\in[c_i,\bar{c_i}]}\sum_{t\in[t_k,\bar{t_k}]} P(t,c).log\frac{P(t,c)}{P(t).P(c)}$
Mutual information	$log\frac{P(t_k,c_i)}{P(t_k).P(c_i)}$
Chi-square	$\frac{\|T_r\|.[P(t_k,c_i).P(\bar{t_k},\bar{c_i})-P(t_k,\bar{c_i}).P(\bar{t_k},c_i)]^2}{P(t_k).P(\bar{t_k}).P(c_i).P(c_i)}$
NGL coefficient	$\frac{\sqrt{\|T_r\|}.[P(t_k,c_i).P(\bar{t_k},\bar{c_i})-P(t_k,\bar{c_i}).P(\bar{t_k},c_i)]}{\sqrt{P(t_k).P(\bar{t_k}).P(c_i).P(\bar{c_i})}}$
Relevancy score	$log\frac{P(t_k\|c_i)+d}{P(\bar{t_i}\|\bar{c_i})+d}$
Odds ratio	$\frac{P(t_k\|c_i).(1-P(t_k\|\bar{c_i}))}{(1-P(t_k,c_i)).P(t_k\|\bar{c_i})}$
GSS coefficient	$P(t_k,c_i).P(\bar{t_k},\bar{c_i}) - P(t_k,\bar{c_i}).P(\bar{t_k},c_i)$

Table 3.1 – *The summary of the mathematical definition of the different term selection functions related to the task of text classification. Here, probabilities are interpreted on an event space of documents T_r (e.g. $P(t_k, c_i)$ is the probability of term t_k occurring in a random document x which has the category c_i.*

et al. 1995), NGL coefficient (Ng et al. 1997), Information Gain (Caropreso et al. 2001), Mutual Information (Dumais et al. 1998), Relevance Score (Wiener et al. 1995), and GSS coefficient (Galavotti 2000). Table 3.1 gives a summary of the mathematical definition of these term selection functions.

Even though the task is to find similar texts, the use of term selection has increased the performance of various NLP tasks due to the different domains, type of corpora, language etc. Due to these differences, choosing one particular term selection function for an NLP task should be based on comparisons between the different methods which could be a long and exhaustive process (Sebastiani 2002). Because of this, a general linguistic criteria is used to select terms which is widely used in NLP. The linguistic criteria considers all the words present in the corpus to be terms except for the function words. This is because the function words do not play a major part in representing the text in semantic spaces or in other words describing the content of the text.

Term weights

Weighting Terms

As term selection is an important part, giving suitable weights to these terms is important as well (Buckley 1993). Term weights act as a medium to provide information on how useful a term is to relate texts. This information is represented by some numerical property of the term itself for example the number of times the term is present in the text. There are different weighting schemes available among which the most common and popular is the tf-idf weighting scheme (Salton and McGill 1986). It

combines local and global weights of a term, trying to get an overall view on how important the term is with respect to a particular text as well as in a collection. The tf-idf weighting scheme, w, for a term is given in equation 3.1.

$$w(term) = tf(term) * idf(term) \tag{3.1}$$

where, tf is the local weight that corresponds to the number of times a term occurs within a text and idf is the global weight which corresponds to the logarithmic value of the ratio between its document frequency and the total number of texts in the collection (Jones 1972). There are many variants of this tf-idf model including the SMART weighting systems (Salton and Buckley 1988), Okapi-BM25 weighting system Robertson et al. (1996), INQUERY weighting system (Broglio et al. 1994), and the delta variant of SMART and BM25 scheme (Georgios Paltoglou 2010). The SMART, BM25 and their delta invariant weighting functions are shown in tables 3.2, 3.3, and 3.4. These tables show the weighting system of a term which has 3 parts, the local weight, global weight, and the normalization factor. The product of these three parts, as with the tf*idf method, give the weights for terms. The local weight gives some value to a term that shows how much does it represent the text with respect to other terms in the text whereas, the global weight gives a value that shows how much does the term represents the text with respect to the other text in a collection. The normalization factor penalizes the term weights for the length of the text in accordance to its length.

Notation	Term frequency
n (natural)	tf
l (logarithm)	$1 + (logtf)$
a (augmented)	$0.5 + \frac{0.5tf}{max_t(tf)}$
b(boolean)	$1 \quad if > 0, \quad else \quad 0$
L (log ave)	$\frac{1+log(tf)}{1+log(avg.dl)}$
o (BM25)	$\frac{(k_1+1).tf}{k_1((1-b)+b.\frac{dl}{avg.dl}+tf}$

Table 3.2 – *SMART notation for term frequency variants.* ***maxt(tf)*** *is the maximum frequency of any term in the document and* ***avg.dl*** *is the average document length with respect to the number of terms. For ease of reference, we also include the BM25 tf scheme. The k_1 and b parameters of BM25 are set to their default values of 1.2 and 0.95 respectively (Jones et al., 2000)*

Claveau (2012) has shown that BM25 performs better than the tf*idf method in the tasks for information retrieval, text mining, and segmentation. In the third Text Retrieval Conference (TREC-3), the probabilistic weighting scheme BM25 and INQUERY performed better than the SMART weighting systems (Singhal et al. 1996) and as shown by Georgios Paltoglou (2010), the delta variant of the SMART and BM25 weighting scheme perform better than the invariant one whereas, some have found them inconsistent (Manning et al. 2008). The normalization factor in these weighting methods plays a significant role especially in retrieval systems (Singhal et al. 1996) but it has been seen not to be useful

Notation	Inverse Document Frequency
n (no)	1
t(idf)	$log\frac{N}{df}$
p (prob idf)	$log\frac{N-df}{df}$
k (BM25 idf)	$log\frac{N-df+0.5}{df+0.5}$
$\triangle$(t) (Delta idf)	$log\frac{N_1.df_2}{N_2.df_1}$
$\triangle$(t') (Delta smoothed idf)	$log\frac{N_1.df_2+0.5}{N_2.df_1+0.5}$
$\triangle$(p) (Delta prob idf)	$log\frac{(N_1-df_1).df_2}{(N_2-df_2}$
$\triangle$(p') (Delta smoothed prob idf)	$log\frac{(N_1-df_1).df_2+0.5}{(N_2-df_2}.df_1+0.5$
$\triangle$(k) (Delta BM25 idf)	$log\frac{(N_1-df_1+0.5).df_2+0.5}{(N_2-df_2+0.5}.df_1+0.5$

Table 3.3 – *SMART notation for inverse document frequency variants. For ease of reference we also include the BM25 idf factor and also present the extensions of the original formulations with their $\triangle$ variants.*

Notation	Normalization
n (none)	1
c (cosine)	$\frac{1}{\sqrt{w_1^2+w_2^2+...+w_n^2}}$

Table 3.4 – *SMART normalization where w_i is the weight of the term i.*

in other applications such as in collecting answers to frequently asked questions (Manning et al. 2008).

This shows that like the term selection problem, the selection of a weighting function is difficult and has to be studied properly for each task and method. For instance, Nakov et al. (2001) and Dumais (1991) studied the effects of several weights used by Latent Semantic Analysis (LSA), which is a text representation method, to solve the text categorization and information retrieval tasks respectively. They show that using proper weights, the performance of the text representation can be improved compared to the baseline tf*idf weights. The weights that they used are listed in table 3.5 and 3.6.

Type	Local weights
term binary	$L(i,j) = 1if\ tf(i,j) > 0$ else 0
term frequency	$L(i,j) = tf(i,j)$
logarithmic	$L(i,j) = log(tf(i,j)+1)$

Table 3.5 – *The local weights concerning the term frequency tf of term i in document j.*

Similar to LSA, there are several other methods that are used to represent texts. All of these methods require terms and their weights to help represent text. In the next sections, we present Vector Space Model (VSM), Latent Semantic Analysis (LSA), Principle Component Analysis (PCA) and Independent Component Analysis (ICA) text representation methods.

Type	Global weights
normalization	$G(i) = 1/sqrt(\sum_j L(i,j)^2)$
gfidf	$G(i) = gf(i)/df(i)$
idf	$G(i) = 1 + \log(\text{ ndocs } /df(i))$
global entropy	$G(i) = 1 + \sum_j p(i,j) \log p(i,j) / \log ndocs$
entropy	$G(i) = H(d\|i) = \check{} \sum_j p(i,j) \log p(i,j)$

Table 3.6 – *The global weights concerning the term i document j where, gf is the global frequency, df is the frequency of document, ndocs is the total number of documents and $p(i,j) = tf(i,j)/gf(i)$*

3.1.2 Vector Space Model

Vector Space Model

Vector Space Model (VSM) is a method of representing texts as vectors in a common vector space using the bag of words approach. In 1979, Salton was the first to present this model for the purpose of Information Retrieval (IR) (Salton 1979) to find documents in a pool of documents that are related to a given query. It was used in the System for the Mechanical Analysis and Retrieval of Text (SMART) (Dubin 2004). In IR, once the texts and queries are represented as vectors, a similarity metric is used to find the similarity values between them and according to this value a ranked list of texts is generated. The similarity metrics are explained in detail in Section 3.2

Text Vectors

In VSM, the texts of a corpus are represented as vectors, called text vectors, using the terms and its weights chosen by one of the methods presented in section 3.1.1. Each text vector will have as many dimensions as there are terms in the corpus and each dimension corresponds to a particular term with values equal to its weight. The set of vectors representing the texts form a $M \times N$ matrix where, M is the number of terms and N is the number of texts represented.

Any text unit can be represented in this model. Salton (1979) used documents but other text units like sentences or text segments can be used as well. Let us consider the following example in which a term-sentence matrix is created from a corpus of three sentences and how the sentences are represented in the vector space using terms.

S1: The elephants are in the zoo.
S2: Tigers are small compared to elephants.
S3: Tigers are also in the zoo.

We take the words elephants, zoo, and tigers as our terms and create the term-sentence matrix as in table 3.7 where the term weight is a binary value indicating the presence with 1 and absence with 0.

Figure 3.1 shows the projection of each sentence in the vector space. The representation of text in the vector space, created from term selection and weighting functions, gives a different perspective on how texts are viewed. Using VSM, closeness between text has been possible to rank unlike while using boolean methods, whose main idea is using logical

Boolean Model

operators namely AND, OR, and NOT between texts (Manning et al. 2008).

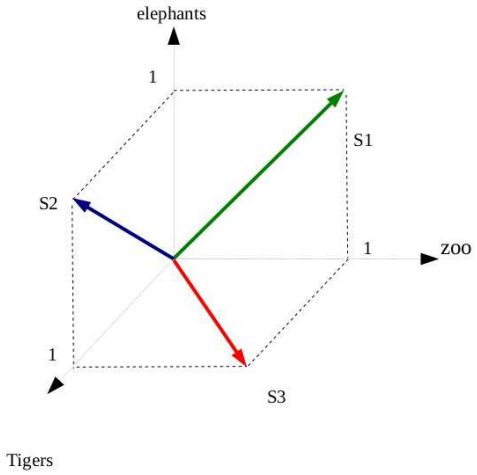


Figure 3.1 – *The projection of the three sentences S1, S2, and S3 in the vector space.*

In the vector space of VSM, where the terms represent the dimensions, the terms are assumed to be linearly independent from each other. This implies that the terms have no relation to each other in anyway for instance, the co-occurrence between terms or semantic relation such as synonymy. This pairwise orthogonality assumption is a disadvantage because dependencies between terms do exist which VSM is unable to exploit and creates problems as mentioned by Wong et al. (1987). They present a new method called Generalized Vector Space Model (GVSM), which introduced term to term correlations which deprecated the pairwise orthogonality assumption. Tsatsaronis and Panagiotopoulou (2009) included semantic information of synonymy and polysemy using WordNet within the model of GVSM to include semantic relatedness. As GVSM, another extension of the VSM which do not assume pairwise orthogonality is the Topic-based Vector Space Model (TVSM) (Becker and Kuropka 2003) where documents are represented by the term vectors whose dimensions are pre-selected topics. An extension on TVSM is the enhanced TVSM (eTVSM) (Santos et al. 2012) which finds relations between terms using some ontology.

The comparisons between these extended models of VSM has been performed on different applications and mostly all the methods shows some performance enhancement while comparing to VSM. GVSM was applied to document retrieval and showed that it is more effective than VSM but is computationally quite intense (Wong et al. 1987). On the other hand, the enhanced version of the GVSM was used for TREC and

	s1	s2	s3
elephants	1	1	0
Tigers	0	1	1
zoo	1	0	1

Table 3.7 – *The term sentence matrix*

showed slight improvements (Tsatsaronis and Panagiotopoulou 2009). Similarly, eTVSM showed slight improvements on spam filtering where as TVSM has only been theoretically compared with VSM Igor12. Despite the slight improvements using these representations, the extra effort and computation required has made VSM a popular choice representing texts.

Rather than extending the representation of VSM, there has been efforts to extract more information from the VSM model for text representation. In the following section we present three such methods that try to extract more information from VSM. One of them is Latent Semantic Analysis (LSA) which is presented in the next section.

3.1.3 Latent Semantic Analysis

"*Latent Semantic Analysis (LSA) is a theory and method for extracting and representing the contextual usage meaning of words by statistical computations applied to a large corpus of text*" as stated by Landauer and Dumais (1997). Many applications that use LSA to represent texts have outperformed applications using VSM. LSA uses a mathematical analysis call Singular Value Decomposition (SVD) to identify a linear subspace in the VSM space that captures most of the variance in the collection of text. This approach is believed to extract the *latent semantics* of texts. The latent semantics is the hidden information present in the text which cannot be represented only using surface level contingencies produced by term weights. LSA has been used in solving problems that deal with word sense disambiguation (Pino and Eskenazi 2009), information retrieval (Deerwester et al. 1990), text segmentation (Choi et al. 2001) and so on.

LSA method represents texts in two steps. The first step is to represent texts as vectors to create a matrix as in VSM, e.g. term-text matrix or any term-context matrix, which is usually a rectangular matrix. Next, SVD is used on this matrix which is a form of factor analysis and is well-known in linear algebra. SVD decomposes the rectangular matrix, $\mathbf{M}$, into three other matrices.

$$\mathbf{M} = \mathbf{TSD}^T$$

where, $\mathbf{T}$ and $\mathbf{D}$ are column-orthogonal matrices and $\mathbf{S}$ is a diagonal $m \times m$ matrix which contains m singular values of $\mathbf{M}$ such that the singular values are in the descending order, $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_m$. The largest singular values are chosen such that $k \ll m$ and the three matrices are multiplied to reconstruct the approximated vector space :

$$\mathbf{M} \simeq \mathbf{M_k} = \mathbf{T_k S_k D_k}^T$$

This matrix $\mathbf{M_k}$ is a re-composed matrix of the original matrix $\mathbf{M}$ which is presented in the fig 3.2. The aim of selecting a set of singular values is to capture the most important structure but also reducing noise and variability (Berry et al. 1995).

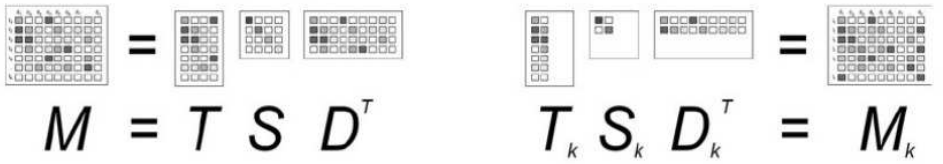


Figure 3.2 – *Singular Value Decomposition with and without selecting a set of singular values.*

The effectiveness of LSA boils down to two main factors. The first is the weights which populate the initial matrix given to LSA to extract the latent semantics and the other is the decision of the optimum number of singular values to select. There have been researches on the performance of different weights (Nakov et al. 2001) and the selection of the singular values (Wild et al. 2005, Landauer et al. 1998, Deerwester et al. 1990) but on specific task and corpus which indicates that the selection of the weight function as well as the number of singular values should be done after experiments on the task in hand. For example, Nakov et al. (2001) made experiments on document categorization and have listed some local and global weights showing that these weights are important and influence the results. The list of local and global weights that they experimented with are presented in table 3.5 and 3.6. The local weights using $log(tf(t,d)+1)$ and global weights using entropy were the weights that performed well with 15 singular values in their experiments. The weight functions work as the tf-idf function where the weight is computed by multiplying the first part which is the local weight, shown in table 3.5, and the second part which is the global weight, shown in table 3.6. Using different local and global weights, the effectiveness of LSA could be measured. As the performance of methods depend on the term and weighting function selection, choosing one could be a tricky task and should be empirical.

Wild et al. (2005) on the other hand presented a number of heuristics that could be used to select the singular values:

- *Percentage of cumulated singular values (share)*: Using a normalized vector, selected singular values are those highest singular values whose sum (divided by the sum of all singular values) equals a specified percent or share.
- *Absolute value of cumulated singular values (ndocs)*: The selected singular values have a sum greater or equal to the number of contexts.
- *Fraction of number of terms*: The number of selected singular values are a fraction of the total indexed terms, usually 1/30 or 1/50.
- *Fixed number of factors*: The number of selected singular values are explicitly given e.g. 10 factors. This number has to be determined depending on the text corpus.
- *Kaiser-Criteria* : The singular values are selected according to the Kaiser-criteria which states that the singular values having a value greater than 1 are selected.

LSA has proved practically useful in extracting the semantics to some extent, but the theoretical foundations remain incomplete and hard to support. A statistical version of LSA called Probabilistic Latent Semantic Analysis (PLSA) is presented by Hofmann (1999) and is based on mixture decomposition derived from a latent class model. PLSA has a sound statistical foundation and defines a generative model of the data. Even though PLSA uses probabilistic modelling of text, it does not take into consideration the probabilistic model between texts or documents. Unlike PLSA, Blei et al. (2003) presented the Latent Dirichlet Allocation (LDA) which is a generative probabilistic model and takes into consideration the document level probabilities while assuming each document is a mixture of latent topics.

Probabilistic Latent Semantic Analysis

Latent Dirichlet Allocation

PLSA and LDA may show better performance than LSA in some NLP tasks, such as information retrieval (Hofmann 1999) and paraphrase detection (Guo and Diab 2012). On the other hand, there are cases where LSA perform better, for instance, in the task of sentiment analysis (Maas et al. 2011). Even with this inconsistency and the incompleteness issues that LSA may have (Landauer et al. 1998), LSA is a much simpler model without any need of training and has shown to have a descent performance. Similar to LSA, another method that tries to extract information from the VSM model to represent texts is Principle Component Analysis (PCA) and is explained in the next section.

3.1.4 Principle Component Analysis

PCA

Principal component analysis (PCA) is a statistical method which uses orthogonal transformation to find patterns in high-dimensional data which are called principle components and are linearly uncorrelated. These components are able to highlight the data's similarities and differences. Since patterns in data can be hard to find in data of high dimension, PCA is a powerful tool for analysing data. PCA has been used in many fields such as image retrieval (Jolliffe 2002), document retrieval, text categorization and summarization (Vikas et al. 2008).

PCA is similar to LSA in the sense that it takes as input the text vector representation as in VSM and it may use Singular Value Decomposition (SVD) to extract features. PCA can also be performed by the eigenvalue decomposition of a covariance matrix. PCA is usually performed using SVD rather than covariance matrix because it is computationally efficient (Jolliffe 2002) and tends to be numerically accurate[1]. PCA extracts principle components which are the dimensions of the space on to which the texts are positioned and the distance between them represent how closely they are related. These principle components are orthogonal to each other and if the data or in our case the text vectors follow the Gaussian distribution, the components are guaranteed to be independent to each other. This will then make a better representation of the text in the

[1] R prcomp package documentation:
http://stat.ethz.ch/R-manual/R-patched/library/stats/html/prcomp.html

space.

PCA takes two steps to represent texts, the first is using SVD to extract the principle components and the second step is to map the texts on to the space represented by the principle components. Here are the steps to perform PCA:

1 SVD takes in the text representation created by VSM, $\mathbf{M}$, which is a rectangular matrix whose rows are normalized by subtracting the mean of each row. SVD decomposes the matrix $\mathbf{M} = \mathbf{TSD}^T$ into three other matrices as explained in section 3.1.3.

2 The matrix $\mathbf{T}$ is the eigenvectors of the covariance matrix $\mathbf{M}^T\mathbf{M}$ and the matrix $\mathbf{D}$ is the eigenvectors of the matrix $\mathbf{MM}^T$.

3 The PCA transformation that preserves the dimensionality is given by $\mathbf{Y}^T = \mathbf{M}^T\mathbf{T}$ and the closeness between texts is derived from this transformation of the space.

There are other eigenvector-based multivariate analyses among which PCA is the simplest. It extracts features of texts from the original space represented by VSM. These features form a subspace whose dimensions correspond to the maximum-variance directions in the original space. Texts when mapped on to this subspace will highlight their similarity and differences. This helps finding the similarity between texts. Similar to PCA and LSA, another method which extracts features from a given set of texts to represent them mathematically that has shown encouraging results. This method is the Independent Component Analysis (ICA) and is explained in the following section.

3.1.5 Independent Component Analysis

ICA

Independent component analysis (ICA) is a statistical method used to discover hidden features from a set of non-Gaussian measurements or observed data such that the sources are maximally independent. ICA has been traditionally used in signal processing and its intuition can be illustrated using the classical blind signal separation example of the cocktail party (Honkela et al. 2010). In this example, two people stands in a room and speak simultaneously. Two microphones that are placed in two different places will each record a particular linear combination of the two voices. These recorded signals are mixtures of the speech signals with different proportions depending on the relative distance of the recorder to each sound source. ICA is then used to separate the original speech signals which are the underlying features from the observed mixtures as shown in figure 3.4. Considering text as mixtures of some underlying features, we could use ICA to extract them to represent the text using these features with the hypothesis that the text from which the features are extracted follow the non-Gaussian properties.

In many problems, we assume the normality of the distribution making it a Gaussian Distribution. However, there are many situations where

Gaussianity does not hold for instance in natural signal based on sensory organs such as in the cocktail party problem where speech signals are involved, electrical signals from different brain areas, and natural images. Even though text formation is not "natural" in the same sense because it is mostly an encoding process, the complexity of the language phenomenon justifies it to be treated as a stochastic process. Whether the texts are governed by the Gaussian distribution or not is still up for debate. But the sparseness of words with large probability mass for values close to zero with heavy tails as mentioned in Zipf's Law seem to make the distribution of words non-Gaussian.

If Gaussianity was assumed in the cocktail party problem, we could perform a Principal Component Analysis (PCA) or a Factorial Analysis (FA). The resulting components would be two new orderly voice combinations and fail to isolate each speaker's voice as shown in figure 3.3. PCA finds projections which have maximum variance whereas ICA finds projections which are maximally non-Gaussian and is able to extract the two different voice signal from the mixture of the two signal making ICA superior to PCA in finding underlying factors in a non-Gaussian scenario (Hyvärinen 1999).

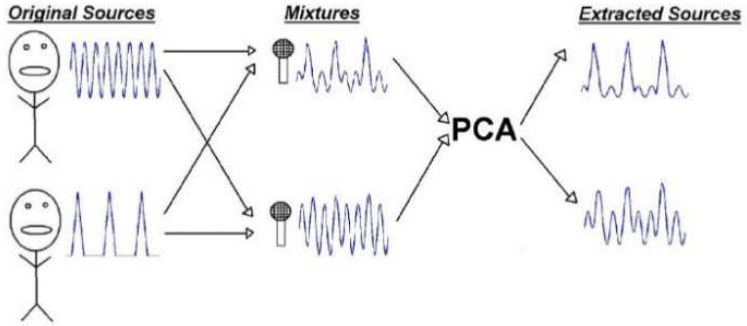


Figure 3.3 – *PCA used in the cocktail party problem to extract the underlying features.*

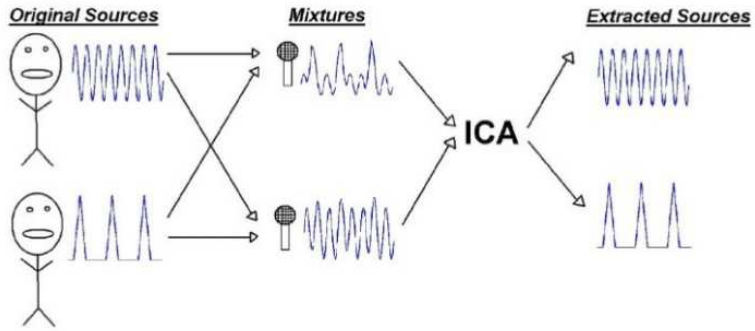


Figure 3.4 – *ICA used in the cocktail party problem to extract the underlying features.*

There are two main differences between ICA and PCA which also illustrates ICA's property. First, in ICA, there is no order of magnitude associated with each component. In other words, there is no better or worst components (unless the user decides to order them following his own criteria). Second, the extracted components are invariant to the sign

of the sources. For example, in image processing, a white letter on a black background is the same as a black letter on a white background.

In the classic version of the linear ICA model (Jutten Herault 1991; Common 1994; Hyvarinen et al. 2001), each observed random variable $x = (x1, x2, ..., xn)^T$ is represented as a weighted sum of independent random variables $s = (s1, ..., sk, ..., xn)^T$, such as

$$x = As \tag{3.2}$$

where A is the mixing matrix that contains the weights which are assumed to be different for each observed variable and s is the vector of the independent components also called the latent variables or sources. If we denote the columns of matrix A by a_i the model can be written as :

$$x = \sum_{i=1}^{n} a_i s_i \tag{3.3}$$

The statistical model in equation 3.2 is called the ICA model which describes how the observed data are generated by a process of mixing the components s_i. Both the mixing matrix A and the independent components s are learned in an unsupervised manner from the observed data x. The observed random variable in our case could be the frequency of a word in a particular context and the independent variables refer to the underlying variables.

The starting point for ICA is the assumption that the components s_i are statistically independent. ICA can be seen as an extension to principal component analysis (PCA) and factor analysis. The main difference between ICA and PCA is, while PCA finds projections which have maximum variance, ICA finds projections which are maximally non-Gaussian. PCA is useful as a pre-processing technique that can reduce the dimension of the data with minimum mean squares error. In contrast, the purpose of ICA is not dimension reduction. ICA is computed in a stochastic manner and the time complexity cannot be directly stated. However, the convergence of the Fast-ICA algorithm (Hyvarinen 1999) is cubic, which makes *Fast-ICA* it feasible to use with real applications. For the FastICA algorithm where the data matrix X is considered to be a linear combination of independent components,

$$x = AS \tag{3.4}$$

where columns of S contain the independent components and A is a linear mixing matrix. The dimension of the data is first reduced by PCA in order to decorrelate the data, to reduce over-learning and to get the square mixing matrix A. After variance normalisation, n independent components which create a feature representation in the component space are extracted with ICA.

In terms of text, the observed data x is a m-by-n term-text matrix where columns represent text and rows represent terms as done by VSM. All the

terms, which are selected using term selection process, present in a collection of text are used for the building of the x matrix and the columns are all the text in the collection. Each element corresponding to a particular term and text in the matrix is the weight given to that particular term with respect to that text as mentioned in section 3.1.1. Fast-ICA is applied on x matrix producing the mixture matrix A and the independent components S by assuming the independence of the rows in S. The independent components in matrix S gives the ICA feature representation for text. Each component s_k encodes some interesting features extracted from the m text. Using these new features or components which are orthogonal to each other we project our term-text matrix in this space. For instance, $\vec{i}$ be a text vector of a given text i, then the projection of this vector in the new space will be :

$$i_p = i^T \times S \tag{3.5}$$

Projecting all the text in this new space, we get a new vector representation of the text x matrix.

ICA is able to reduce the redundancy in the data, extract underlying features, and find interesting projections (Hyvärinen 1999). These properties of ICA make it a good candidate to project the VSM representation of text on to new projections or dimensions that is able to represent the text with new features rather than just terms. Representation of text using ICA is a good idea if the assumption that the text follows a non-Gaussian distribution is true. Honkela et al. (2010) have shown some indication that the distribution of words do tend to show non-Gaussian properties and ICA could be helpful in extracting hidden word features. This gives a good basis for the assumption on text and on the representation of text using ICA.

3.2 Automatic Alignment and Similarity Measures

Alignment is the process in which text that are close or similar to each other are connected and the process to achieve this automatically is presented in this section. In section 2.2, we mentioned two different types of alignments, i.e., Pair-wise alignment and Group-wise alignment. In the Pair-wise alignments, all the possible text pairs that can be generated from a corpus are considered to be aligned or not whereas in the Group-wise alignment, texts in a corpus are grouped into clusters in such a way that each cluster consists of text which are related.

Both of these alignment processes consist of two steps. The first step in both of these alignment process is the same, where a similarity value is assigned to every possible pair of texts. Section 3.1 explains how texts are represented as text vectors which determine their positions within a space represented with different dimensions. Each dimension in the text vector indicates some property of the text and its value determines how much of that property is present in the text. The closeness of two texts in the space determines how closely they relate to each other. This closeness is computed using similarity measures which find the distance between

the vectors and generate a value of closeness.

The second step of the alignment process, in the pairwise alignment method, is to estimate a threshold value. The text pairs whose similarity value is above or equal to this threshold are considered similar and the rest are not. The thresholds in alignment tasks are usually done experimentally against an already annotated data but there are methods to estimate the threshold analytically such as the Probability Thresholding Principle (Lewis 1995). This *probability thresholding* is possible only in the presence of a theoretical result that indicates how to compute the threshold that maximizes the expected value of the effectiveness function which is a special case in IR, therefore in most of the cases, experimental methods are commonly used to determine the threshold. The basic steps for experimental methods are :

1 Calculate the similarity between all the possible text in the training corpus

2 Rank them

3 Calculate the effectiveness measure at every position of the rank

4 Find the level at which the effectiveness measure is optimal

5 Select the level as the threshold

The effectiveness measure in the steps for the experimental method could be recall, precision, or f-measure depending on the application. These effectiveness measures are also the evaluation methods which are explained in detail in section 4.3. Other methods like logarithmic regression models may also be used to estimate the threshold (Nelken and Shieber 2006). Using these experimental methods where thresholds are estimated on the annotated training data, the assumption is that the training data represents the testing data which is difficult to assure. The training data has to be selected properly depending on the available data to ensure that it is as close as possible. Usually, training data are selected randomly from a set whereas, if multiple sets are present then equal number of elements from the different sets are randomly selected. This random selection tries to insure the representativeness of the training set. In a large scale real life scenario, it is difficult to decide on the number of sets present in the collection so the usual way is to select randomly.

Unlike the pairwise alignment method, the second step of the clustering alignment method would be clustering the text using clustering algorithms which use the pairwise similarity values. The clustering algorithms are presented in section 4.2.4. The following section will present various similarity measures that give a value to each text pair.

Similarity Calculation

The representations of text as vectors provide a mathematical basis for assigning continuous values to the amount of similarity between two texts

using different corpus based similarity measures. Similarity measures give a value to how similar two texts are to each other by giving a value to the distance between the vectors of the text are independent of the terms and weights scheme used in the text space. There are two main properties (Huang 2008) that a similarity metric must satisfy which are listed below:

Properties of a metric

a) Distance must be symmetric, which means the distance between x to y and y to x should be the same or, d(x,y) = d(y,x)

b) The similarity metric should satisfy the triangle inequality or, d(x,z) $\leq$ d(x,y) + d(y,z)

In the remaining part, we present different similarity measures that are used in IR which may or may not satisfy the properties of the similarity metric. These similarity measures are explained on the basis of term vectors but can be applied to any vectors that represent the text.

Cosine Similarity

There are several similarity measures that have been proposed among which cosine similarity is very popular. Cosine similarity between two texts is based on the angular difference between the vectors of the two texts represented in the VSM. The calculation of the angular difference between two term vectors $\vec{t_a}$ and $\vec{t_b}$ is :

$$Sim(\vec{t_a}, \vec{t_b}) = \frac{\vec{t_a}.\vec{t_b}}{|\vec{t_a}| \times |\vec{t_b}|} \tag{3.6}$$

The range of the similarity value can be from -1 to 1 but the range can be and is usually made in the range of [0,1] by normalizing the text vector where 1 is the upper bound indicating the two texts are identical while 0 indicates the text have nothing in common. The important property that cosine has is the independence of document length. The cosine similarity between documents $\vec{t_a}$ and another document $\vec{t_b}$ which is the combination of two identical document $\vec{t_a}$ will give a value of 1 and the similarity of these documents with any other document, l, will be the same e.g. d($\vec{t_a}$,l) = d($\vec{t_a}$,l). Even though Cosine similarity measure is popular in the field of NLP to find similarity between texts, it does not satisfy the triangle inequality which does not make it a true metric (Korenius et al. 2007).

Euclidean Distance

Unlike cosine similarity measure, Euclidean distance is a true similarity metric to find difference between vectors. This measure satisfies the similarity metric properties. Given two documents d_a and d_b represented by their term vectors $\vec{t_a}$ and $\vec{t_b}$ is defined as

$$Sim(\vec{t_a}, \vec{t_b}) = (\sum_{t=1}^{m}(w_{t,a} - w_{t,b})^2)^{1/2} \tag{3.7}$$

where the term set is $T = t_1, .., t_m$. The range of value given by this measure do not have an upper bound and depends on the values in the

vectors. A value of 0 indicates the text are identical which is also the lower bound of this metric. The normalization of euclidean distance called the chord distance which produces the range of value from 0 to 2, where 0 indicates identical text and 2 indicates that the text has nothing in common (Korenius et al. 2006). The chord distance can be calculated as :

$$Sim(\vec{t_a}, \vec{t_b}) = (\sum_{t=1}^{m}(\frac{w_{t,a}}{(\sum_{t=1}^{m} w_{t,a}^2)^{1/2}} - \frac{w_{t,b}}{(\sum_{t=1}^{m} w_{t,b}^2)^{1/2}})^2)^{1/2} \tag{3.8}$$

The normalization done by dividing the square root of the sum of the square of the vector minimizes the effect of large values which are easily affected in the euclidean distance. Even though cosine similarity is not a true metric, euclidean distance and cosine similarity have similar performance as mentioned by Korenius et al. (2007).

Jaccard coefficient

Jaccard coefficient is also known as Tanimoto coefficient and also another metric which satisfies the similarity metric property (Huang 2008). The coefficient of two texts represented by $\vec{t_a}$ and $\vec{t_b}$ is calculated as given below:

$$Sim(\vec{t_a}, \vec{t_b}) = \frac{\vec{t_a}.\vec{t_b}}{|\vec{t_a}|^2 + |\vec{t_b}|^2 - \vec{t_a}.\vec{t_b}} \tag{3.9}$$

The range given by Jaccard coefficient is 0 to 1 where 0 meaning that the text are not similar and 1 indicating the text are identical. This coefficient finds the value based on the number of intersection divided by the union of the terms in the text.

Dice coefficient

Dice coefficient is a similarity measure over sets. It is named after Lee Raymond Dice, and is also referred to as Sorensen-Dice's coefficient (Lewis et al. 2006). The coefficient of two texts represented by $\vec{t_a}$ and $\vec{t_b}$ is calculated as given below:

$$Sim(\vec{t_a}, \vec{t_b}) = \frac{2|\vec{t_a}.\vec{t_b}|}{|\vec{t_a}|^2 + |\vec{t_b}|^2} \tag{3.10}$$

The range given by this similarity measure is 0 to 1 where 0 meaning that the text are not similar and 1 indicating the text are identical. This similarity measure is not a proper distance metric because it does not possess the property of triangle inequality. Even though it is not a metric it has been used in IR.

Kullback-Leibler Distance

Kullback-Leibler(KL) distance is the symmetric version of Kullback-Leibler divergence which considers text as a probability distribution of texts. KL divergence finds the difference between the probability distribution of text to give a value for similarity. For two distributions P and K,

the KL divergence on a finite set X is shown in (3.11).

$$D_{KL}(P\|K) = \sum_{x \in X} P(x) log \frac{P(x)}{K(x)} \tag{3.11}$$

This measure is not symmetric but there exists symmetric versions. In Pinto et al. (2007), some of the symmetric versions are presented and shown that there is not much difference between them. One of the symmetric versions of KL divergence is to select the max of the KL divergence as in (3.12).

$$D_{KLD} = max(D_{KL}(P\|K), D_{KL}(K\|P)) \tag{3.12}$$

Similarly to choosing the maximum of the two KL divergence there is another version which uses the average between the two KL divergence. The equation 3.13 shows the calculation of the distribution of the terms in the vocabulary, V.

$$P(t_k, d_i) = \begin{cases} \beta * P(t_k|d_i), & \text{if term } t_k \text{ occurs in the document } d_i \\ \epsilon, & \text{otherwise} \end{cases} \tag{3.13}$$

where,

$$P(t_k|d_i) = \frac{w(t_k, d_j)}{\sum\limits_{t_k \in d_i} w(t_k, d_i)} \tag{3.14}$$

and

$$\beta = 1 - \sum_{t_k \in V, t_k \notin d_i} \epsilon \qquad \text{such that,} \qquad \sum_{t_k \in d_i} \beta * P(t_k|d_i) + \sum_{t_k \in V, t_k \notin d_i} \epsilon = 1 \tag{3.15}$$

Here, $w(t_k, d_i)$ is the weight of the term k in document d_i. The computation of the probability consists of a smoothing model based on back-off. A small value from the weight of the terms present in the document are deducted as shown in equation 3.15 and a portion, ϵ, is given to the terms that are not present as shown in equation 3.13.

Compared to KL divergence, there is another famous method to find similarity between probability distributions called Jensen-Shannon and is based on it, but is symmetric and is always a finite value (Manning and Schütze 1999).

Conclusions

This chapter presents four different methods which are able to represent texts as vectors and similarity measures that are able to find the distance between these vectors to give a value on the closeness between texts. These representation methods and similarity measures are explained in this chapter which are important parts of the automatic alignment process.

Automatic alignment is the process of creating links between texts based on similarity. This is a two step process among which the first step is common among the two different types of alignment, i.e. Pair-wise

alignment and Group-wise alignment. In this common step the similarity between texts are calculated on the basis of vectors representing the text using similarity measures. These text representations are built on terms, that are the foundation of the context of texts, and the weights that are assigned to each of them. The selection of terms and their assignment of weights are important aspects because they are used by the text representation methods and in turn have a significant effect on their performance. The four different methods try to represent texts by placing them in a space of a fix dimension. Each of these dimensions is a feature of the text and are used by similarity measure to find texts that are close to it. There are many similarity measures, some are a true metric whereas some are not but are still useful and are used in the field of IR.

The second step in the automatic alignment is the alignment part. For the pair-wise alignment process, a threshold on the similarity value between texts is determined. The texts having similarity values greater or equal to this threshold is considered to be aligned. Whereas, for the group-wise alignment process, the similarity values are used by clustering algorithms to cluster the texts in to aligned groups.

In the next chapter, we present the state of the art methods for text segmentation, the alignment process and the evaluation of these methods.

Text Segmentation and Short text Alignment

4

"Don't Panic."

-Douglas Adams, *The Hitchhiker's Guide to the Galaxy*

Contents

In chapter 2, an overview about multimodal monolingual corpus alignment and the three sub-processes that constitute this alignment process is explained. These sub-processes are segmentation of text, representation of these segments and finally the text alignment itself. Text representation has been covered in detail in chapter 3. In this chapter, we focus on the state-of-the-art of the segmentation and the text alignment processes. We also present methods that will evaluate the alignment process.

4.1 Segmentation

4.1.1 Using Lexical Cohesion

Lexical Cohesion is the lexical relationship within a text that holds the *Lexical Cohesion*

text together and gives its meaning (Halliday and Hasan 1976a). These properties have been exploited in order to find thematic shifts within text which in turn provides segment boundary information. Two of the lexical properties namely word reiteration and co-occurrence are used in different segmentation methods. In this section, we discuss four different types of segmentation methods which use word reiteration, clustering methods that directly or indirectly use word reiteration, co-occurrence, and statistical methods that exploit lexical cohesion.

Word Reiteration

TextTiling

In a text, some words are used multiple times and this information has been useful to some extent to find segment boundaries. TextTiling (Hearst 1997) uses this distribution of words to give a similarity measure between two adjacent blocks, where each block consists of a fix number of windows of terms. This similarity measure is used to segment the text considering each window gap to be a candidate segment boundary. The process starts by tokenizing the text in adjacent windows of 20 terms and forming blocks each made of 6 consecutive windows. Similarity between the two adjacent blocks, b_1 and b_2, is computed using the cosine similarity measure shown in equation 4.1:

$$sim(b_1, b_2) = \frac{b_1.b_2}{|b_1||b_2|} = \frac{\sum w_{b_1}(t).w_{b_2}(t)}{\sqrt{\sum w_{b_1}^2 \sum w_{b_2}^2}} \tag{4.1}$$

where, b_1 and b_2 are the blocks of windows between which the similarity is to be measured, w is the weight for the term, t, which in this case is the frequency of the term in the window. Once the similarity is calculated, the blocks are shifted one window at a time to find the similarity between every window gap. Using this similarity measure, the depth score is assigned to each window gap. The depth score indicates how strong the subtopic change is present and is based on the distance from peaks on both sides of the valley to that valley as illustrated in Figure 4.1.

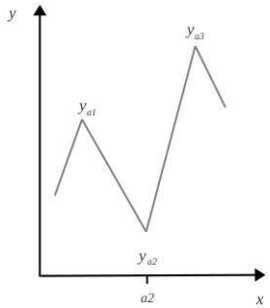


Figure 4.1 – *The y and x axis represents the similarity value and the window gap respectively. The depth score at gap a2 is* $(y_{a1} - y_{a2}) + (y_{a3} - y_{a2})$.

Once this depth score is computed for every gap, the scores are sorted and used to determine the segment boundaries. The number of boundaries are automatically determined using the values of standard deviation and

variance of the distribution of the depth scores

Galley et al. (2003) uses word reiteration to make chains called lexical chain. A chain is linearly created, starting at the beginning of the text, and is connected to the same word repeated along the text. For every new word seen, a new chain is created. These chains are divided into sub-chains when there is a long hiatus of h consecutive sentences with no occurrence of the term. Once these chains are created, the cosine similarity between two windows, i.e. two sentences A and B, are calculated using the weights that are determined by the lexical chain overlap as shown in equation 4.2 where, R_i is the lexical chain of term, t_i, present in the text. L_i is the length[1] of the lexical chain of the term t_i and L is the length of the text. Quite similarly to the TextTiling method, the depth score at every local minimum, m_i, is computed and the number of segments are calculated. *Lexical Chain*

$$cosine(A, B) = \frac{\sum_i w_{i,A}.w_{i,B}}{\sqrt{\sum_i w_{i,A}^2 . \sum_i w_{i,B}^2}} \tag{4.2}$$

where,

$$w_{i,T} = \begin{cases} score(R_i) & \text{if } R_i \text{ overlaps } T \in \{A, B\} \\ 0 & \text{otherwise} \end{cases}$$

$$scoreR_i = freq(R_i).log(\frac{L}{L_i})$$

Here, the depth score at a point is computed by finding maxima of cohesion on its left and right side, l and r respectively, and is called the hypothesized segmentation probability:

$$p(m_i) = \frac{1}{2}[LCF(l) + LCF(r) - 2.LCF(m_i)] \tag{4.3}$$

where, $LCF(x)$ is the lexical cohesion function, which is calculated by equation 4.2, at some gap x.

There are other methods which use the windowing technique. One of them is Dotplotting (Reynar 1994). In this method, the word reiteration property is exploited using the concept of overlapping of words. It is a graphical method for detecting topic boundaries, of which segmentation of text is done. A Dotplot is basically a binary repetition matrix where the axis have the same text and each unit of the axis corresponds to the term position within the text. A dotplot of a four Wall street journal articles concatenated is shown in Figure 4.2. We can observe that there are three visible rectangular structures which indicate the boundary for segmentation. An improvement on this Dotplotting method was proposed by Ye et al. (2006) which use a sentence-based lexical similarity rather than the word-based similarity. *Dotplotting*

Stokes et al. (2002) used word reiteration in the sense of concept reiteration as explained by Halliday and Hasan (1976a). The method exploits

[1] All lengths are measured in number of sentences

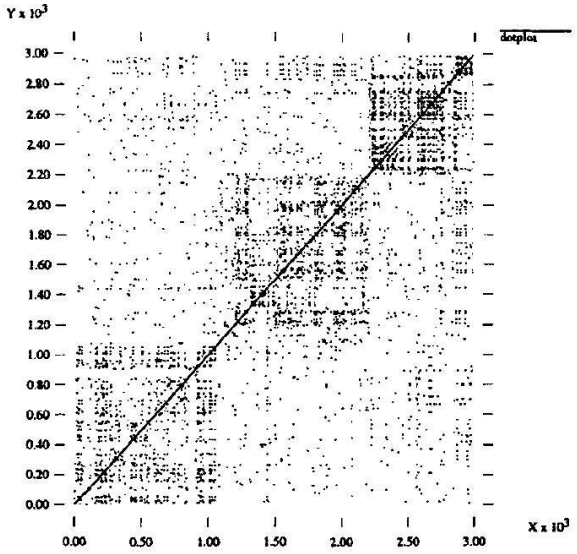


Figure 4.2 – *A Dotplot of four concatenated Wall Street journal articles (Reynar 1994).*

synonymity along with word repetition, word association through specialization/generalization, word association through part-whole/whole-part relationships for segmenting CNN transcripts of the broadcast news. Using these information from WordNet a lexical chain is created connecting words. Firstly, the words are given a part of speech tag which will help find relations in the WordNet. After this step, a chain is created starting from the first word. The chain continues towards the next word, by adding the word to the chain, if the above mentioned lexical cohesion is satisfied else a new chain is started. This process is a single pass clustering algorithm. To prevent weakly cohesive chains, some criteria are considered which take into account the distance between consecutive words forming the chains and the path length in the WordNet taxonomy. Once all the words in the text are in chains, the segment boundaries are detected. The boundaries are detected by the hypothesis that states "A high concentration of chain begin and end points exist on the boundary between two distinct news stories." The number of chains that start and end at each sentence boundary is computed and the mean of these value is calculated. This mean is the threshold above which all the sentence boundary is a segment boundary.

Clustering

Segmentation can also be done using clustering algorithms. The clustering algorithms presented here are directly or indirectly dependent on word reiteration. The objective of clustering is to group things together to make a collection whereas segmentation in the general term is to divide something into small things but if we think of segmentation as a grouping problem then we could group smaller units into a collection of units and consider each collection being a segment.

C99

Choi (2000) has presented a linear text segmentation algorithm named C99 for segmenting a document. This method uses a clustering process

based on word reiteration for segmentation. It has three steps:

1. Creating a similarity matrix of sentences
2. Creating a rank matrix from the similarity matrix
3. Clustering sentences using the ranking matrix

The similarity matrix is created by computing the similarity between two sentences using the cosine similarity measure where the weights are the frequencies of the terms present in the sentence. However as stated by the author, this similarity measure is unreliable, thus a rank matrix is created from this similarity matrix. The rank matrix is created by replacing its rank in a local region. The rank is the number of neighbouring elements with a lower similarity value. This is demonstrated in Figure 4.3.

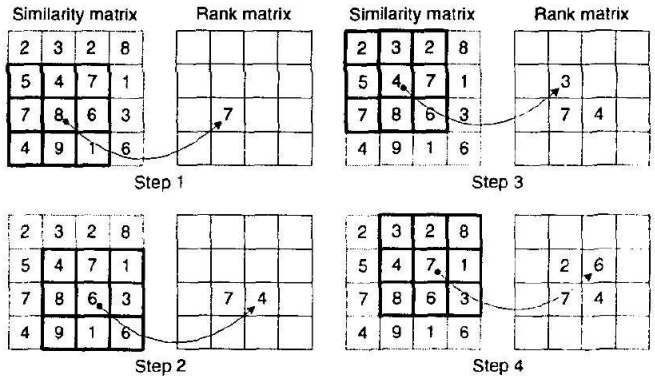


Figure 4.3 – *A working example of the creation of the rank matrix using similarity matrix with a 3*3 rank mask.*

In this work, a 11*11 rank mask was used for the segmentation process. Each rank is normalized using the following equation,

r = No. of elements with a lower value / No. of elements examined

Clustering is done once the rank matrix is created. The clustering process is based on Reynar's maximization algorithm (Reynar 1998). For the segmentation, a text segment is defined by two sentences i,j which represent the region along the diagonal of the rank matrix. $B = b_1, ..., b_m$ is a list of m coherent text segments of a document. $s_{i,j}$ is the sum of the rank values in a segment and $a_{i,j} = (j - i + 1)^2$ be the inside area. Initially, the entire document is placed in B as one coherent text segment. At each step of the process, one of the segments in B splits in such a way that D, the inside density of the rank matrix is maximized where,

$$D = \frac{\sum_{k=1}^{m} s_k}{\sum_{k=1}^{m} a_k} \tag{4.4}$$

where, s_k and a_k refers to the sum of rank and area of segment k in B. An improvement on this method has been made by Choi et al. (2001), in which the similarity matrix is created using Latent Semantic Analysis (LSA) rather than the cosine measure. They have shown that the LSA

metric is twice as accurate as the cosine matrix.

Lamprier et al. (2008) on the other hand computes the similarity between sentences which are derived from a global representation of words rather than just frequency within the sentences as in the previous method. First, they represent the text into a matrix corresponding to the vectorial model (Baeza-yates and Ribeiro-Neto 1999) in which each word is represented by a vector of weights according to the document it is present in. Using this matrix, sentences are clustered into groups using the cosine similarity measure with a variant of the single-pass clustering method by Frakes and Baeza-Yates (1992). Once the initial clusters have been created, re-clustering is done and sentences from one clusters are moved to another cluster with respect to the classes of their neighbourhood to create collection of sentences having the same topic. The score, S, with which one sentence is moved from one cluster to the other is given by the following equation:

$$S(C_k, U_i) = Sim(C_k, C_{c_1}) + \beta \times S(C_k, U_{i-1} + \beta \times S(C_k, U_{i+1}) \qquad (4.5)$$

where $Sim(C_k, C_{ci})$ represents the similarity between the class C_k and the class C_{ci} that contains the sentence U_i and β is a constant above 0 which is determined empirically. Once this re-clustering converges, the boundaries are placed between two sentences that are not present in the same class.

Yaari (1997), unlike the methods mentioned above, works at the paragraph level and used a modified version of the Hierarchical Agglomerative Clustering (HAC) algorithm where paragraphs are clustered to find boundaries. He finds the structure of the text by clustering these paragraph units. The HAC has been modified to keep the structure of the text intact by clustering only adjacent units at a time rather than any units based on the cosine similarity between them.

Co-occurrence

Co-occurrence is a type of collocation where two words co-occur more often than would be expected by chance and these words are assumed to be interdependent. Co-occurrence has been given a notion of "relatedness" by Halliday and Hasan (1976a) signifying that knowledge of words can be gathered from the words which with they co-occur.

Ferret (2007) uses the sliding window technique as in the TextTiling method for text segmentation but differs from it as the method is based on co-occurrence and topic. The topic is extracted from the shared nearest neighbour (SNN) graph which is built from the similarity graph made from the co-occurring words in the text. These topics are the subset of the terms that are present in the text. Ferret uses two lexical cohesion measures between the windows. These lexical cohesions are computed with equation 4.6 and 4.7:

$$LC_{rec}(x) = \frac{2 \cdot card(W_l \bigcap W_r)}{card(W_l) + card(W_r)} \tag{4.6}$$

where $LC_{rec}(x)$ is the lexical cohesion at the point x between the two windows. W_l and W_r are the words on the left and right window.

$$LC_{top}(x) = \frac{card(TW_l) + card(TW_r)}{card(W_l) + card(W_r)} \tag{4.7}$$

where,

$$TW_{i \in l,r} = (W_i \bigcap T_w) - (W_l \bigcap W_r) \tag{4.8}$$

and T_w is all the topics that has been identified. The sum of these two lexical cohesions is the depth score between the two windows. The segmentation is done similarly to the TextTiling method.

Kaufmann (2000) also use co-occurrence for segmentation but this co-occurrence is extracted using the help of external knowledge. The segmentation process follows the windowing method, similar to the TextTiling method, but with different weights for the cosine similarity measure and is called the VecTile algorithm. In this method, Kaufmann computes the co-occurrence of a term and a keyword. A dictionary is present from which the possible terms are selected and the keywords are a set of meaningful words which may not be present in the dictionary. Kaufmann represented the weight, w_i, of a term, t_i, as a $|K|$-dimensional vector, $w_i = n < t_i, k_i > k_{i\epsilon\{0,...,n\}}$ representing the number of occurrences of the term, t_i, and the keyword, k_i, co-occurring within a certain window of 35 words. To perform the similarity between two windows, the vector of all the terms present in each window are added to get a vector that represents its window and the similarity between them are derived from the cosine similarity measure. Once the similarity value is computed the segmentation process is followed as in the TextTiling method.

Caillet et al. (2004), unlike other methods mentioned in this section, uses a clustering algorithm for segmentation, where the basic units are paragraphs. They used clustering twice, for concept learning, which is defined as a cluster of terms formed according to the co-occurrence of words, and for paragraph clustering. First, each term is represented as a n-dimensional vector, $w =< n(w,i) > i\epsilon\{0,...,n-1\}$ representing the number of occurrences of w in each paragraph x_i. Using this vector, terms are clustered together using the x-mean clustering method according to the Bayesian Information Criterion (BIC). According to the clusters, the paragraphs are represented in the $|C|$-dimension vector, $x_i =< n'(c,i) > c\epsilon\{1,...,|C|\}$, where the feature $n'(c,i)$ means the number of occurrences of terms from cluster c in the paragraph. They use classification maximum likelihood (CML) approach along with these vectors for segmentation.

Statistical Methods

Utiyama and Isahara (2001) presented a statistical method for text segmentation using a unigram language model. The evaluation of the ability

of this language model provides the computation of the lexical cohesion of a segment. Their method consists of two important steps which includes the estimation of the language model δ_i and the computation of the generalized probability of words in the segment S_i.

The unigram language model δ_i, of a segment S_i, specifies a distribution over all the words of the text that is to be segmented. This language model is computed using the Laplace smoothing (Manning and Schütze 1999), by

$$\delta_i = \left\{ P_i(u) = \frac{C_i(u)+1}{n_i + K}, \forall u \in V_K \right\} \tag{4.9}$$

where V_K is the vocabulary of the text of size K and $C_i(u)$ the count of word u in S_i with n_i number of words. The generalized probability of words in S_i is computed with this language model δ_i as a measure of lexical cohesion using

$$\ln P[S_i; \delta_i] = \sum_{j=1}^{n_i} \ln P[w_j^i; \delta_i], \tag{4.10}$$

where w_j^i represents the j^{th} word in S_i. The method then searches the segmentation that are the most probable segmentation of a sequence of basic units of words or sentences $W_{a_i}^{b_i}$ in segment S_i among all possible m segmentations on a text with n words given by,

$$\hat{S} = \arg\max_{S_1^m} \sum_{i=1}^{m} (\ln(P[W_{a_i}^{b_i}|S_i]) - \alpha \ln(n)), \tag{4.11}$$

where $P[W_{a_i}^{b_i}|S_i]$ denotes the generalized probability of the sequence of basic units in S_i and α is the parameter that controls the importance between lexical cohesion and segment lengths.

Guinaudeau et al. (2012) present improvements on this method in two ways. The first incorporates different kinds of additional information to the lexical cohesion measure, i.e. incorporating the confidence measures given by automatic speech recognition systems, using semantic relations, to improve the generalized probability measure of lexical cohesion and the second one uses language model interpolation techniques to provide better language model estimates.

4.1.2 Using Discourse cues

Discourse cues are clues that are present in a text that help to understand the text. It helps understand the text because cues in the text from the same source are consistent and gives different hints about the intention of the speaker for example, when a person gives emphasis on a word, we know it is important. Discourse cues are mostly useful in segmenting audio and video transcripts. In segmentation, some discourse cues can be indications of segment boundaries by giving clues because they indicate how information is transferred such as:

- broadcasts start and end with the anchor

- reporter segments are preceded by an introductory anchor segment and together they form a single story
- commercials serve as story boundaries and so on.

Maybury (1998) uses discourse cues for finding segment boundaries in broadcast news. In the paper, CNN Prime News were analysed to find different types of cues as the following:

- Start of Broadcast : "Good evening, I'm Kathleen Kennedy, sitting in for joie chen."
- Anchor-to-Reporter Handoff : "We're joined by CNN's Charles Zewe in New Orleans. Charles?"
- Reporter-to-Anchor Handoff: "Charles zewe, CNN, New Orleans."
- Cataphoric Segment: "Still ahead on Prime news"
- Broadcast End: " That wraps up this monday edition of Prime news"

All these discourse cues with other knowledge like multimedia cues (e.g. detected silence, black or logo keyframes) and temporal knowledge (e.g. in CNN Prime News Programs, the weather segments are on average 18 minutes after the start of the news) are used to form a Finite state machine which segments the text.

Maybury has used discourse cues that are specific only to CNN Prime News. These discourse cues might not be able to segment text from other broadcast news because the cues chosen are too specific to one particular broadcast news and the structure in which that news is presented. In contrast, Passonneau and Litman (1993) have used general linguistic discourse cues namely referential noun phrase, cue words such as "now", and pauses. The segmentation process is evaluated on a corpus of spontaneous, narrative monologues. Each linguistic cue is used independently in a hand crafted rule based algorithm. This algorithm has shown that the boundaries detected were comparable to boundaries identified by humans with regards to recall, but the precision was much lower.

4.1.3 Using Hybrid System

Galley et al. (2003) use lexical cohesion and discourse cues in a probabilistic classifier to determine the position of the segment boundary. The lexical cohesion is based on word repetition. The discourse cues used are silence, cue phrases, overlapping of speech, and speaker change. These features are selected and combined using the C4.5 algorithm and C4.5 rules (Quinlan 1993) to learn rules for segmentation. This process is claimed to be useful for both text as well as audio transcripts and has shown that this hybrid system outperforms systems that use only lexical cohesion or discourse cues.

Beeferman et al. (1999) uses probability for the segmentation process where the probability models the existence of a boundary for a segment

at each sentence boundary. The probability depends on two types of features which have been extracted incrementally from a pool of features using an exponential model. The two classes of features are topicality and cue-word features. Topicality feature uses long and short distance language models detecting topic change in the text where as the cue-word features are words which may be domain specific that are usually present near the segment boundary. There are other methods (Blei and Moreno 2001, Yamron et al. 1998) that probabilistically model the topic as one of the component of Beeferman as mentioned above. These topic models are generally built from a large set of training data and even though they increase the precision of segmentation they also restrict their scope.

In this section, the state of the art methods for segmentation is presented for the creation of text units which is one of the prerequisite for the alignment process. The different varieties of texts make each presented method perform differently as each of them uses different techniques for segmentation. The similarity criteria for each method is also different creating different text segments. It is difficult to compare all the segmentation methods and choose the best one, as one method which performs better on one text may not perform as good on the other (Galley et al. 2003). Though, in most of the cases, the statistical methods are one of the best performing methods which accounts for high variability in segments length (Guinaudeau et al. 2012). Among the state of the art methods, TextTiling and C99 are the methods that are mostly used as the baseline to compare the performance of new methods.

4.2 Monolingual Short Text Alignment

Aligning text pairs is the process of organizing text in such a way that two text segments of text are indicated as similar. This alignment of text pairs can be done manually or automatically. A good example of a manually annotated corpus is the METER corpus (Gaizauskas et al. 2001) which was built to measure the reuse of texts. The texts in the METER corpus consists of newspaper articles and are manually collected and classified by a professional journalist with some annotations at the phrasal or even lexical level. The manual process as mentioned in chapter 2 is a cumbersome process because each text pair have to be checked for similarities taking a huge amount of human effort and time. This has lead to automatic text pair alignment. The PAN-PC-09 corpus (Barrón-Cedeño et al. 2010), used to detect plagiarism, was created automatically but in a controlled environment. The text were artificially synthesized in a way that would resemble plagiarism within the text. This control over artificially generated text makes a good annotation as there is a prior knowledge of which text is plagiarised. This control also restricts the corpus to resemble naturally occurring plagiarised text and may represent only a subset of the problem of plagiarism which is a drawback of artificially created text and their annotations.

There has been little research done on the automatic alignment of naturally occurring text pairs. Among them, we will be presenting exist-

ing works of automatic text alignment methods, which are closely related to our aim, proposed by Barzilay and Elhadad (2003), and Nelken and Shieber (2006), in which monolingual alignment has been done at the sentence level and by Hatzivassiloglou et al. (2001), where the alignment of text segments are done for the application of summarization. Paraphrase alignment on the sentence level is a more specific type of textual alignment and could be considered as a sub-problem of text alignment in general. We present the works from Mihalcea and Corley (2006), Barzilay and McKeown (2001), and Islam and Inkpen (2008). Most of them give a general method to measure textual similarity and evaluate their methods on the task of sentential paraphrase alignment.

4.2.1 Sentence Alignment

We start by mentioning the work of Barzilay and Elhadad (2003). In this work the sentence alignment is done between two collections from the Encyclopedia Britannica and Britannica Elementary. The algorithm for alignment consists of 4 steps as follows:

1. Vertical Paragraph Clustering
2. Horizontal Paragraph Mapping
3. Macro Alignment
4. Micro Alignments

1) Vertical Paragraph Clustering

The text in the two corpora are segmented into physical structures of paragraph. In this first step, the paragraphs are clustered together for each collection. This clustering is done by the hierarchical complete link clustering based on the cosine measure similarity given by the word overlap of the paragraphs. The objective of this clustering is to group together paragraphs that convey similar information and for that they have ignored function words and replaced text specific attributes, such as proper names, dates and numbers, by generic tags.

2) Horizontal Paragraph Mapping

Once the paragraphs are clustered, they have used this information of clusters to learn the mapping from one cluster to the other in the other corpus. This learning process is done using training data of manually aligned sentences from one corpus to the other and if at least one sentence is mapped then they assume the paragraphs are mapped. Using this training data, the mapping between two paragraphs is learned using the BoosTexter classification tool. This tool also takes into consideration the cosine similarity that uses word overlap but without the replacement of text specific attributes between the two paragraphs and the paragraph number of the paragraph in question. With this the training part is completed.

3) Macro Alignment

This step is the start of the alignment process. Given two texts to be aligned, sentences that have a high cosine similarity measure are aligned

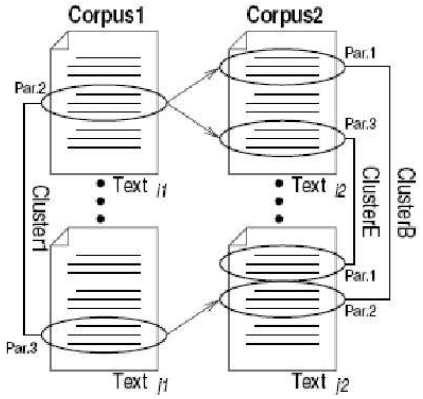


Figure 4.4 – *Training set for the paragraph mapping step where each arrow indicates at least one sentence is aligned between the clusters.*

and for the rest of the paragraphs in each text they are clustered as in the step 1, Vertical Clustering. Once each paragraph is clustered in its group, Each paragraph in one text is mapped to the paragraphs on the other text using the rules learned from step 2, Horizontal Paragraph Mapping.

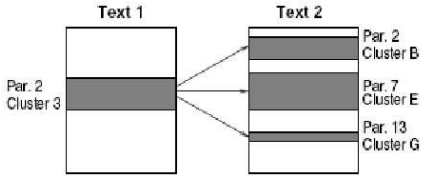


Figure 4.5 – *Macro Alignment between the paragraph in Text1 with the candidate paragraphs of Text2.*

4) Micro Alignments

In this step, the sentences of the aligned paragraphs are aligned using the local similarity and the optimal alignment weight using dynamic programming in such a way that the context of the sentence is taken into account. Using this algorithm to align the Encyclopedia Britannica and Britannica Elementary they have been able to increase the precision with previous models like SimFinder (Hatzivassiloglou et al. 2001) and standard Cosine measure as shown in the Table 4.1.

System	Precision
SimFinder	24%
Cosine	57.9%
Full Method	76.9%

Table 4.1 – *Precision at 55.8% Recall*

A better precision for this alignment between the Encyclopedia Britannica and the Britannica Elementary has been achieved by Nelken and Shieber (2006) who presented a different algorithm for alignment. This algorithm also has a learning process and uses dynamic programming

which takes into account the context while finding the similarity as done by Barzilay and Elhadad (2003). But the learning process and dynamic programming are very different to the work of Barzilay et al.
Their algorithm has three parts:

1. Finding the similarity score between two sentences.
2. Learning the probability of two sentences being matched, using the similarity score.
3. Using dynamic programming for alignment, using the learned probability.

Unlike in the work of R. Barzilay et al., R. Nelken et al. do not find mapping between paragraphs for the sentence alignment. They work at the sentence level and find the similarity between them using the cosine similarity measure with the weights as given below:

$$w_s(t) =_{def} TF_s(t)\dot{l}og(\frac{N}{DF(t)}) \tag{4.12}$$

where, t is a term in sentence s, $TFs(t)$ is a binary indicator of whether t occurs in s, and $DF(t)$ is the number of sentences among all the other sentences, N, in which t occurs. Using this similarity score, the probability of two sentences being matched given a similarity score is learned and has been shown to follow a sigmoid-shaped curve as shown in the Figure 4.6.

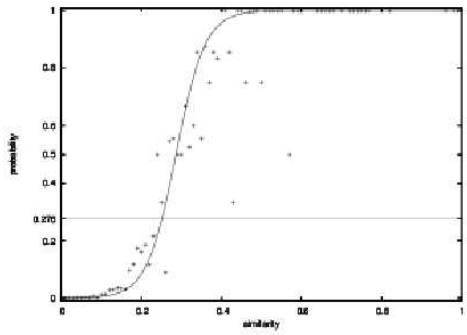


Figure 4.6 – *Logistic Regression for Britannica training data where, the Y-axis represents the probability whereas the X-axis represents the similarity value.*

A regression model representing this shape is used to find the probability of match w.r.t. each similarity value which is given below:

$$p = \frac{e^{a+bx}}{1+e^{a+bx}}$$

where, a and b are parameters learned from the corpora. In addition to this, they also set a threshold for the probability distribution above which the sentences are considered as a match. Using this regression model alone for alignment has yield competitive accuracy compared to Barzilay's algorithm. The accuracy is further improved using a global alignment dynamic programming algorithm, which prunes many spurious matches. In the step of global alignment using dynamic programming, instead of the similarity measure it's corresponding probability of match is used.

While aligning sentences from one corpus to the other, the context is used and a global alignment is made. The importance of context has been illustrated both by Barzilay and Elhadad (2003) and Nelken and Shieber (2006) in their paper. The results are shown in the Table 4.2.

Algorithm	Precision
SimFinder(Hatzivassiloglou et al. 2001)	24%
Word Overlap	57.9%
Barzilay & Elhadad	76.9%
Nelken et al. with TF*IDF	77.0%
Nelken et al. with TF*IDF + Align	83.1%

Table 4.2 – *Precision of different algorithms to align at 55.8% recall*

Other than aligning the Encyclopedia Britannica and Britannica Elementary, they have applied this algorithm to align gospels but using a different set of regression parameters and threshold. The result for the precision is shown in the Figure 4.7.

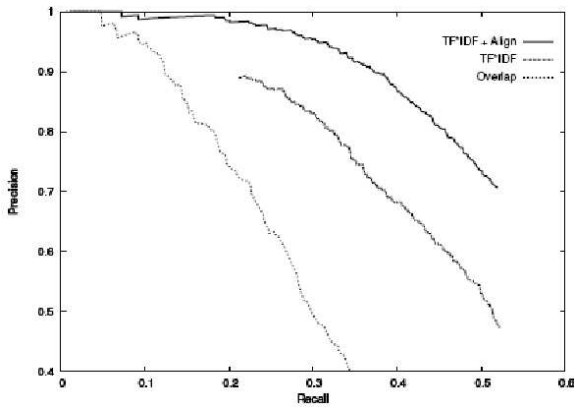


Figure 4.7 – *Precision/Recall curves for the gospels.*

4.2.2 Paraphrase Alignment

Sentential paraphrase alignment is a specific type of text alignment of two sentences which are alternate ways of conveying the same information. These sentences are the ideal example of similar texts as they convey the same information. In general, two texts that have at least one information in common are considered as similar which makes paraphrase alignment a part of the alignment problem. In this section, we will present a general overview of the works from Mihalcea and Corley (2006), Barzilay and McKeown (2001), and Islam and Inkpen (2008).

Barzilay and McKeown (2001) follows the methodology developed for Machine Translation (MT). In the corpus, sentences are first aligned using an unsupervised learning algorithm. After the alignment, paraphrases are extracted based on the assumption that aligned sentences with similar context are paraphrases. Sentences of some corpus are aligned to each other based on dynamic programming (Gale and Church 1991) whose

weight functions are based on the number of common words in the sentence pair. Once the sentences are aligned, noun and verb phrases are identified using part-of-speech tagger and chunker (Mikheev 1997). The process first starts with the identical words in the aligned sentences which act as the seed for an incremental learning process of good contexts and use them to learn new paraphrases.

Their model is based on the DLCoTrain algorithm proposed by Collins and Singer (1999) which applies a co-training procedure to decision list classifiers for two independent sets of features. One set of features describes the paraphrase pair itself, and another set of features corresponds to contexts in which paraphrases occur. These features are use to extract the new set of positive and negative paraphrasing examples. This method was evaluated against the agreement with two judges. The first judge selected the paraphrase without the context with a kappa value of 0.68 whereas the second judge selected the paraphrase with context information with a kappa of 0.97. These experiments were done on monolingual parallel corpora which was created by collecting multiple English translation of the same source text which gives advantages in learning the context rules.

Unlike Barzilay and McKeown (2001), Mihalcea and Corley (2006) have proposed a method for measuring the semantic similarity of texts. The method computes the similarity using the corpus-based and knowledge-based similarity measures between the words the texts consist. Given two texts T_1 and T_2, the similarity between them is determined using the scoring function given in equation 4.13.

$$sim(T_1, T_2) = \frac{1}{2}\left(\frac{\sum_{w \in T_1}(maxSim(w, T_2) * idf(w))}{\sum_{w \in T_1} idf(w)} + \frac{\sum_{w \in T_2}(maxSim(w, T_1) * idf(w))}{\sum_{w \in T_2} idf(w)}\right) \tag{4.13}$$

In this similarity measure, the *maxSim* is a function to find the similarity score of words such that the value is between 0 and 1. Along with the word similarity, the measure also takes into account the *word specificity* so that higher weights are given to two specific words (e.g. *collie* and *sheepdog*), and give less importance to the similarity measured between generic concepts (e.g. *get* and *become*). The specificity of the word is determined by the inverse document frequency (idf) and is calculated from the British National Corpus (BNC). The *maxSim* are determined using corpus-based and knowledge based measures between words that have the same part-of-speech. One of the reasons behind this is that most of the knowledge based measures cannot be applied across part-of-speech.

The two corpus based metrics used to compute the similarity between words are Latent Semantic Analysis (LSA) (Landauer et al. 1998) (Section 3.1.3) and Point-wise Mutual Information (PMI-IR) (Turney 2001). The PMI-IR of two words w_1 and w_2 is calculated as in equation 4.14.

$$PMI\text{–}IR(w_1, w_2) = log_2 \frac{hits(w_1\ AND\ w_2)}{hits(w_1) * hits(w_2)} \tag{4.14}$$

Here, $hits(x)$ be the number of documents retrieved in which x is present when the query x is given to AltaVista search engine. Another highly related similarity measure based search engine is the Google Similarity Distance (Chilibrasi and Vitanyi 2007) and could be used instead of PMI-IR. Once these metrics are used to find the similarity between words contained in the texts, they are given to the equation 4.13 to get a similarity value of texts between 0 and 1.

The knowledge based methods used the combination of six semantic similarity measures were which take in two concepts and return a value indicating their semantic relatedness. These six measures are : Leacock and Chodorow (Leacock and Chodorow 1998), Lesk (Lesk 1986), Wu and Palmer (Wu and Palmer 1994), Resnik (Resnik 1995), Lin (Lin 1998b), and Jiang and Conrath (Jiang and Conrath 1997). These type of metrics were used to align paraphrases from the Microsoft Research Paraphrase Corpus (MSRPC). Table 4.3 gives the Precision, Recall and F-score for each of the metric. Even though the Precision and Recall values of these methods are high, the external knowledge source that is required and the use of 6 metrics using word net makes it computationally expensive. On the other hand, the PMI-IR method used the AltaVista search engine's advanced "NEAR" search option which is not used any more. This means PMI-IR cannot be used in the same form in new systems (Islam et al. 2008).

Islam and Inkpen (2008) on the other hand proposed the Semantic Text Similarity (STS) method which determines the similarity between two texts from the semantic and syntactic information they contain. The semantic information is generated from calculating string similarity and semantic word similarity whereas the syntactic information is optional and if used, it is calculated using the common-word order similarity function. Finally, the text similarity is derived by combining string similarity, semantic similarity and common-word order similarity with normalization.

This method is a general way of finding a value for similarity between texts and has been evaluated on the Microsoft paraphrase corpus(Dolan et al. 2004). It performs equally well compared to other methods previously presented. Table 4.3 shows the result of the evaluation in terms of the Precision, Recall and F-score.

4.2.3 Paragraph Alignment

SimFinder (Hatzivassiloglou et al. 2001) is a monolingual paragraph alignment system developed at the Columbia University for text summarization. It aligns paragraphs using a clustering algorithm. This clustering is based on a similarity metric which is a combination of multiple linguistic features. These features are selected using machine learning. The features were of two types, Primitive and Composite. Primitive features are single linguistic characteristic which include word co-occurrence, matching noun phrase, WordNet synonyms, common semantic classes

Metric	Precision	Recall	F-score
Semantic similarity (corpus-based)			
PMI-IR	70.2	95.2	81.0
LSA	69.7	95.2	80.5
STS	**74.7**	89.1	**81.3**
Semantic similarity (knowledge-based)			
J & C	72.2	87.1	79.0
L & C	72.4	87.0	79.0
Lesk	72.4	86.6	78.9
Lin	71.6	88.7	79.2
W & P	70.2	92.1	80.0
Resnik	69.0	96.4	80.4
Combined	69.6	**97.7**	**81.3**

Table 4.3 – *Text similarity for paraphrase identification on the MSRP corpus.*

for verbs, and shared proper nouns. Composite features are features that involve pairs of primitive features with different types of restrictions. Three types of restrictions are mentioned which are: the pair of primitive elements occur in the same order, they occur within a certain distance, and each element of the pair of primitive elements is restricted to a specific primitive as demonstrated in Figure 4.8, 4.9, and 4.10. These three restrictions can be combined to make different composite features.

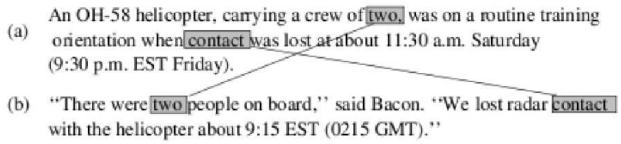


Figure 4.8 – *A composite feature over word primitives with a restriction on order would make the pair of words "two" and "contact" as a match because they have the same relative order.*

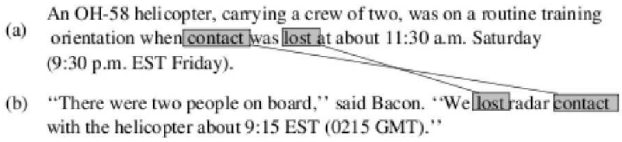


Figure 4.9 – *A composite feature over word primitives with a restriction on distance would match on the pair "lost" and "contact" because they occur within two words of each other in (a) and in (b).*

Originally, 43 features were proposed and among them 11 were selected using RIPPER, which is a rule learning software (Cohen 1996). SimFinder, now, uses 7 features which are selected using a log-linear regression model. This regression model is given below,

$$R = \frac{e^{\eta}}{(1 + e^{\eta})} \tag{4.15}$$

where η is a weighted sum of the features and R is the similarity measure.

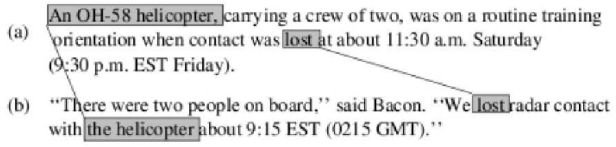


Figure 4.10 – *A composite feature with restrictions on the primitives' type. One primitive must be a simplex noun phrase (in this case, a helicopter) and the other primitive must be a matching verb (in this case,"lost") as the match in (a) and (b).*

The 7 features were selected by an iterative process where at each step different combination of features with different weights are used on this log-linear model. The clustering algorithm used by SimFinder is a non-hierarchical technique , called the exchange method (Spath 1985). With the creation of clusters, the objective of alignment is completed and for the objective of summarization each cluster generates few sentences that are most representative of the cluster which is done by CENTRIFUSER and MULTIGEN (Barzilay et al. 1999) . The alignment process is better than simple $TF * IDF$ and the SMART system as shown in the Table 4.4. In contrast, Barzillay et al. and Nelken et al. has shown that while aligning the Encyclopedia Britannica and Britannica Elementary corpus, their algorithms outperform SIMFINDER as shown by their respective evaluation in Table 4.1 and 4.2.

System	Precision	Recall	F1-measure
Standart TF*IDF	32.6%	39.1%	35.6%
SMART	34.1%	36.7%	35.4%
SIMFINDER	49.3%	52.9%	51%

Table 4.4 – *Evaluation scores for several similarity computation techniques.*

4.2.4 Alignment of Text Clusters

The alignment of text by grouping similar texts together is the task of alignment of text clustering. The text are clustered by organizing the text, for instance making lists, which indicates which texts are grouped together based on a pre-defined criteria. The applications that make use of these type of alignments are summarization, information extraction/retrieval, text categorization and so on. There exist few freely available corpora[1] that deal with text clustering among which are the CICLing-2002 corpus, hep-ex corpus of CERN, and KnCr Corpus from MEDLINE. These corpora consists of abstracts from different domain and were collected for the purpose of text categorization. Like the alignment of text pairs, alignment of text clusters can be done manually or automatically. As manual annotations take time, automatic methods and algorithms of clustering are a viable solution. Here we present some existing algorithms and works on text clustering.

[1]http://users.dsic.upv.es/grupos/nle/?file=kop4.php

Clustering text segments is the task of grouping short texts together into groups in such a way that text segments corresponding to a predefined criteria are found in a unique group. It consists of two steps: the first step is to find the similarity or dissimilarity matrix and then clustering the short texts with the help of this matrix. In this section, we focus on the clustering algorithms used for clustering text segments. There are many clustering algorithms among which K-means and Hierarchical Agglomerative Clustering (HC) algorithms are used the most to cluster text segments. Along the presentation of the clustering algorithm, some existing works on text segment clustering will also be presented.

Clustering algorithms

The hierarchical agglomerative clustering (HC) and Spectral clustering (SPEC), which uses K-means algorithm, methods are described in this section. HC are bottom up algorithms in which elements are merged together to form dendrograms and are used extensively in the field of NLP. Different HC algorithms are present for instance Single Link HC (SHC), Complete Link HC (CHC), Average Link HC (AHC) and so on, but have the same underlying approach and can be formally written as these steps:

1. Compute the dissimilarity matrix.
2. Start with each text segments in one cluster and repeat the following steps until a single cluster is formed :
   (a) Merge the closest two clusters.
   (b) Update the dissimilarity matrix to reflect the dissimilarities between the new cluster and the original clusters.
3. Stop the merging of clusters after the predefined number of clusters are reached.

The main difference between most of the hierarchical clustering algorithm is in step 2a where the closest clusters are determined. Below, we state how closeness is determined for three types of HC algorithms.

Single Link HC (SHC) : This clustering method considers two clusters to be close in terms of the minimum dissimilarities between any two elements in the two clusters.

Complete Link HC (CHC) : This clustering method considers two clusters to be close in terms of the maximum dissimilarities between any two elements in the two clusters.

Average Link HC (CHC) : This clustering methods considers two clusters to be close in terms of the average pairwise dissimilarities of all the pairs of elements in the two clusters.

Along with the HC algorithms Spectral Clustering which are also used in text segment clustering and has been recently used in the community of machine learning (von Luxburg 2007). K-means clustering algorithm is the underlying clustering algorithm of SPEC which is applied on the normalized eigenvectors of the similarity matrix. The algorithm for spectral clustering is given below from (Ng et al. 2001) :

1. Given a set of short texts, $S = \{s_1, ..., s_n\}$, the similarity matrix, $M \in \mathbb{R}^{n \times n}$, is generated using some similarity measures.

2. Create the affinity matrix $A \in \mathbb{R}^{n \times n}$ defined by the Gaussian Similarity function, $A_{ij} = exp(-\|r_i - r_j\|^2/2\sigma^2)$ with $\sigma = 0.5$, if $i \neq j$, and $A_{ii} = 0$, where $r_i, ..., r_j$ are rows of M.

3. Construct the normalized graph Laplacian matrix $L = D^{-1/2}AD^{-1/2}$ where, D is a diagonal matrix whose (i, i)-element is the sum of A's i-th row.

4. Compute the eigenvectors of L and select the k largest eigenvectors and stacking them in columns to form $X = [x_1, x_2, ..., x_k] \in \mathbb{R}^{n \times k}$.

5. Normalize the row's of X to have unit length to form the matrix Y (i.e. $Y_{ij} = X_{ij}/(\sum_j X_{ij}^2)^{1/2}$).

6. Using K-means, cluster the rows of matrix Y into k clusters by treating the row of Y as points in $\mathbb{R}^k$.

Clustering Text Segments

There are few existing methods that cluster text segments and are applied to the text categorization process of clustering abstracts of scientific papers. We will present existing works from Makagonov et al. (2004) and Pinto et al. (2007).

Makagonov et al. (2004) clusters abstracts of two international conferences in order to categorize the papers with only the help of their abstracts using adequate keyword selections and similarity measures. For the keyword selection, they propose two simple criteria based on the relative frequencies of the words in the domain specific documents and the number of documents in which the words are present. On the basis of the selected keywords the similarity between the text segments are calculated. The similarity measure that they propose is the combination of the cosine similarity measure and a polynomial (linear or quadratic) similarity measures (Manning and Schütze 1999). Using this similarity measure on the selected keywords of the abstracts, the dissimilarity matrix is created and is used by two clustering algorithms for the clustering. The algorithms are the K-means clustering algorithm and the nearest neighbour hierarchical clustering algorithm. They have shown that the simple keyword selection creates stable contents of the clusters and that K-means perform better than the hierarchical clustering.

Pinto et al. (2006) also clusters abstracts using the help of one of the three keyword selection process proposed. These keywords are used to create the dissimilarity matrix between abstracts and are used by several hierarchical clustering algorithms to clustering the abstracts. The different keyword selection processes are based on the information about the *document frequency*, i.e. the number of documents that contains the term, *Term Strength*, i.e. probability of how a particular term expresses the text and the *Transition point*, i.e. the value that separates the low

and high frequency words. The selected keywords are used to create the dissimilarity matrix using Jaccard similarity functions, which is used by the k-NN clustering showing that Transition point perform better than the other two keyword selection techniques.

Pinto et al. (2007) uses several symmetric Kullback-Leibler distance to find the dissimilarity between text segments by finding the difference between their distribution. Using this dissimilarity matrix they used four clustering algorithms which includes Single Link HAC, Complete Link HAC, and KStar. They found that the similarity measures do not affect the clustering results significantly.

4.3 Evaluation of Alignments

In the previous sections, we have presented different text representation and the similarity measures that uses this representation and various clustering algorithms mentioned in the section 1.3. Different combination of these systems make different methods for automatic alignments could give interesting results and to judge the performance of each combination we evaluate the results. The general idea of evaluating the different methods is that they are given a set of text to align, called the evaluation set. Within this set, we know which are the actual alignments but are not given to the alignment methods. Once the methods align these text we evaluate the output by giving an evaluation score on the basis of how many of the actual alignments are present in the automatically aligned set and how many are not. We have two types of alignment methods which are evaluated differently and are discussed in the following sections.

4.3.1 Aligned Pairs

The alignment methods that produce pairwise alignments could be evaluated using two views. One view is that the aligned pairs produced by the alignment method are unranked set and the other view is that the aligned pairs produced are ranked list according to the similarity value given to them. For the first view, the most popular evaluation methods are Recall, Precision, and F-measure whereas for the second evaluation method, the Mean Average Precision is used.

Recall, R, is the fraction of the actual aligned pairs which were aligned by the alignment method. Given that a is the text pairs that are aligned by the alignment method and b is the text pairs that are actually aligned, initially present in the evaluation set.

$$R = \frac{\#(a \cap b)}{\#b} \tag{4.16}$$

Precision, P, is the fraction of the aligned pairs that are aligned by the alignment method which are actually aligned.

$$P = \frac{\#(a \cap b)}{\#a} \tag{4.17}$$

Having the two values for Recall and Precision is good because each of them expresses the different aspects of the result. Recall gives the number of actual pairs that could be aligned whereas the precision shows how many of the aligned pairs are actually good ones. The importance of each of these values are important and depends on different circumstances. In general, we would like to have a balance between these two measures which is given by the weighted harmonic mean of Precision and Recall called the **F measure**.

$$Fmeasure = \frac{(\beta^2 + 1)PR}{\beta^2 P + R} \tag{4.18}$$

where, β is a weight in theory could be in the range of [0,∞]. The balanced F measure would give equal weights to Precision and Recall having β = 1 whereas if the β is >1 the preference is more towards precision whereas a value <1 would give recall more preference.

In contrast to evaluating the result of alignment as a set, a different view of evaluating is to take the list of alignments produced by the system which are ranked according to the value of the similarity value called a ranked list. This evaluation is the **Mean Average Precision** (MAP) and is calculated as :

$$MAP(T) = \frac{1}{|T|} \sum_{j=1}^{|T|} \frac{1}{m_j} \sum_{k=1}^{m_j} Precision(R_{jk}) \tag{4.19}$$

where, $|T|$ is the number of aligned texts, m_j is the number of alignments the system provides for the j^{th} aligned text, and $Precision(R_{jk})$ is 0 if the alignments are not found for the j^{th} alignment else 1/r (where r is the rank of the alignment in the rank list produced by the system). MAP tends to evaluate the Precision at all the levels of Recall by doing so weights each text equally in the final reported number.

4.3.2 Aligned Clusters

Clusters can be evaluated using the quality of clusters created which can be measured in two ways. One by using the extrinsic information and the other using intrinsic information. Extrinsic evaluation compares the created cluster against some pre-defined gold standard and determines a value of quality whereas intrinsic evaluation calculates how close elements are in one clusters and how further apart they are in other clusters to determine the quality of the clusters. We will only be using extrinsic measures as it is most commonly used in text categorization.

Clustering F-measure (F) : F is a mapping based measure where evaluation is done by mapping each cluster to a class (Fung et al. 2003) and is based on the principle of IR precision and recall as follows:

$$F(C) = \sum_{C_i \in C} \frac{|C_i|}{S} max_{K_j \in K}\{F(C_i, K_j)\} \tag{4.20}$$

where,

$$Recall(C_i, K_j) = \frac{n_{ij}}{|C_i|} \qquad Precision(C_i, K_j) = \frac{n_{ij}}{|K_j|}$$

and

$$F(C_i, K_j) = \frac{2 \times Recall(C_i, K_j) * Precision(C_i, K_j)}{Recall(C_i, K_j) + Precision(C_i, K_j)}$$

where n_{ij} is the number of items of class C_i present in clusters K_j and S is the total number of items. The F value will be in the range of [0,1], where 1 being the best score.

In Pinto et al. (2007) a slight variation of this method has also been used in clustering short texts which computes the F according to the clusters rather than the class and is computed as

$$F(K) = \sum_{K_j \in K} \frac{|K_j|}{S} max_{C_i \in C}\{F(C_i, K_j)\} \tag{4.21}$$

F-measure computes its value on the overall Precision and Recall which could produce mappings from one class to more than one clusters that are created which may not be a precise evaluation.

Rand Index (RI) and Adjusted Rand Index (ARI) : RI proposed by Rand (1971) compares two clusters using combinatorial approach where they may fall into one of the four categories :

- TP (true positives) = objects belong to one class and one cluster
- FP (false positives) = objects belong to different classes but to the same cluster
- FN (false negatives) = objects belong to the same class but to different clusters
- TN (true negatives) = objects belong to different classes and to different cluster

RI is computed as in 4.22.

$$RI = \frac{TP + TN}{TP + FP + TN + FN} \tag{4.22}$$

Here, TP and TN are considered to be agreements whereas FP and FN are disagreements. The value of RI can range between 0 and 1 where 1 meaning exact match and 0 indicating the maximal difference. The shortcomings of this measure is that the expected value of two random partitions do not take a constant value, for example zero and it gives equal weights to FP and FN.

ARI is an improvement of the Rand Index which is based on counting pairs of elements that are clustered similarly in the classes and clusters (Hubert and Arabie 1985). With the initial setting the ARI can be computed as below:

$$\frac{\sum_{i,j} \binom{n_{ij}}{2} - [\sum_i \binom{n_i}{2} \sum_j \binom{n_j}{2}] / \binom{S}{2}}{1/2[\sum_i \binom{n_i}{2} + \sum_j \binom{n_j}{2}] - [\sum_i \binom{n_i}{2} \sum_j \binom{n_j}{2}] / \binom{S}{2}} \tag{4.23}$$

where n_{ij} is the number of short texts of class C_i present in cluster K_j, n_i is the number of short texts in class C_i, n_j is the number of short texts in the cluster K_j and S is the total number of short texts. The upper

bound of this measure is 1 and corresponds to the best score and the expected value of this measure is zero. Even though ARI overcomes some shortcomings of RI, it also suffers from distributional problems leading to unstable behaviour.

Variation Information (VI) and Normalized Variation Information (NVI) : VI is based on information theory and uses conditional entropy to evaluate clustering (Meila 2007). The value of VI is computed as in 4.24.

$$VI(C,K) = H(C|K) + H(K|C) \tag{4.24}$$

where K is the cluster created by the clustering algorithm and C is the gold standard class. VI measures the distance between the partitions of the same data. The first term $H(C|K)$ measures the amount of information of C that we loose whereas the second term $H(K|C)$ measures the amount of information of K that we gain. The small conditional entropy indicates that the clusters are formed such that the clusters are close to the gold standard. A VI value of 0 indicates that the gold standard and the clusters that are created are exactly the same where as the upper bound indicating the maximal difference between them is $logN$ where N is the number of clusters. VI has many useful properties like the values can be intuitively understood, changes in a cluster doesn't have a global effect and so on (Meila 2007) however, VI is bounded by the maximum number of clusters and the value of VI is heavily dependent on the number of elements being clustered therefore, they are difficult to compare values across different datasets.

VI can be further normalized to NVI (Reichart and Rappopor 2009) so that the range of the evaluation measure is 0 to 1 where 0 indicates and exact match and 1 indicates the maximal difference. NVI is calculated as in equation 4.25.

$$NVI(C,K) = \frac{1}{logN} VI(C,K) \tag{4.25}$$

NVI tries to make the range from 0 to 1 but by doing so it still has problems of comparison between two authors having different number of clusters as mentioned in (Meila 2007).

Validity measure (V-measure) : V-measure is also based on information theory and uses entropy and conditional entropy to evaluate the clusters and overcomes the drawbacks of other information theory evaluation methods (Rosenberg and Hirschberg 2007). The value of V are computed as in (4.26).

$$V = \frac{2hc}{h+c} \quad \text{where,} \quad h = \begin{cases} 1 & H(C) = 0 \\ 1 - \frac{H(C|K)}{H(C)} & \text{else} \end{cases} \quad c = \begin{cases} 1 & H(K) = 0 \\ 1 - \frac{H(K|C)}{H(K)} & \text{else} \end{cases} \tag{4.26}$$

with,

$$H(C) = -\sum_{i=1}^{|C|} \frac{\sum_{j=1}^{|K|} a_{ij}}{S} log \frac{\sum_{j=1}^{|K|} a_{ij}}{S}$$
$$H(K) = -\sum_{j=1}^{|K|} \frac{\sum_{i=1}^{|C|} a_{ij}}{S} log \frac{\sum_{i=1}^{|C|} a_{ij}}{S}$$

$$H(C|K) = -\sum_{j=1}^{|K|}\sum_{i=1}^{|C|}\frac{a_{ij}}{S}log\frac{a_{ij}}{\sum_{i=1}^{|C|}a_{ij}}$$
$$H(K|C) = -\sum_{i=1}^{|C|}\sum_{j=1}^{|K|}\frac{a_{ij}}{S}log\frac{a_{ij}}{\sum_{j=1}^{|K|}a_{ij}}$$

where, a_{ij} is the number of short texts in C_i which is present in K_j. V gives an evaluation score in a range of [0,1], 1 being the best score.

BCubed : In Amigó et al. (2009) a new evaluation metric, BCubed, is proposed which is the only metric which satisfies four intuitive formal constraints that captures the different aspects of the quality of clusters. These constraints are mentioned below:

- Cluster homogeneity : Clusters should contain elements with the same class.
- Cluster completeness: Elements of the same class should be grouped together in a cluster.
- Rag Bag : Introducing disorder into a disordered cluster is less harmful than introducing disorder into a clean cluster.
- Clusters size vs. quantity : A small error in a big cluster should be preferable to a large number of small errors in small clusters.

BCubed evaluates the clusters in terms of the Precision and Recall associated with each item in the distribution. The item precision represents how many items in the same cluster belong to the same category and the item recall represents how many items from its category appear in its cluster. BCubed depends on the correctness of the relation between elements e and e' given that $K(e)$ and $C(e)$ is the cluster and class of the element e. This correctness relation is defined in 4.27.

$$Correctness(e,e') = \begin{cases} 1 \text{ iff } \quad L(e) = L(e') \text{ and } C(e) = C(e') \\ 0 \text{ otherwise} \end{cases} \tag{4.27}$$

This shows that two elements are correctly related when they share the class if and only if they appear in the same cluster. The BCubed Precision and Recall are based on this correctness relation. The BCubed precision of an element is the proportion of correctly related elements in its cluster (including itself). The overall RCubed precision is the averaged precision of all items in the distribution. The BCubed recall is analogous, replacing "cluster" with "class". The BCubed Precision and Recall are given in 4.28 and 4.29.

$$PrecisionBCubed = Avg_e[Avg_{e'.K(e)=K(e')}[Correctness(e,e')]] \tag{4.28}$$

$$RecallBCubed = Avg_e[Avg_{e'.C(e)=C(e')}[Correctness(e,e')]] \tag{4.29}$$

The BCubed F-measure is given as the IR F-measure, which gives an overall balanced measure on the BCubed Recall and Precision.

Conclusions

The textual alignment process can be divided into three sub-tasks which are identifying text units for alignment, representation of these text units and finally the alignment. The state of the art for identification of texts units and the alignment sub-tasks has been explained in this chapter while the state of the art for text representations has been explained in the previous chapter. Along with these two sub-tasks, the state of the art for evaluating the alignments are presented.

As mentioned in chapter 2, the definition of text units mostly depends on the task and it is not always easy to identify them in long texts. There are several methods that has been proposed among which most of them are based on lexical cohesion, discourse cues, or some hybrid systems. Most of these methods are based on topics where each text unit is associated with some topic. The state of the art methods, for identifying the text units, presented in this chapter show that each of them behave differently to different texts as they rely on different techniques making the comparison between them difficult. This in turn makes the task of choosing the best performing segmentation process even more difficult. Even though, statistical methods, in most of the case, are one of the best performing methods which account for high variability in segments length (Guinaudeau et al. 2012), TextTiling and C99 are the methods that are mostly used as the baseline to compare the performance of new methods.

Alignment of text units is based on similarity and it is the arrangement of text units such that two similar text units are some how connected to each other. This arrangement can be done pair-wise or group-wise. In the pair-wise alignment process, two text units are aligned or not depending on their similarity, where as in group-wise alignment the text units that are similar to each other are clustered in a group indicating the alignment. The automatic pair-wise alignment of these text units is presented in this chapter. There has been very few research on aligning naturally occurring text unit pairs. Most of the time the text units are sentences and even much fewer on the level of text segments which could contain more than one sentence. The state of the art on the alignment process of text units show how difficult this task is as the results do not show convincing results.

The group-wise alignment are done using clustering algorithms and we present different hierarchical clustering algorithm and the spectral clustering. Along these clustering algorithms, we present some existing works on short texts clustering but because the evaluation measure used in these experiments were not reliable, the evaluation of the methods are inconclusive. At the end we present the different evaluating methods for pair-wise and group-wise alignment which are accepted and used frequently in practice.

In the next chapter, we present the monolingual multimodal corpus,

how it was created along with alignments. The comparisons between the different text representation methods and similarity measures for the purpose of automatic alignment of text units will be presented empirically.

Building the Gold Corpus

5

Contents

Our attempt in this thesis is to find a method that will be able to align short text segments in a Multimodal Monolingual Comparable (MMC) corpus. The first thing required to achieve this goal is to have such a corpus. In this chapter, we present the methods used to create this corpus and its manual annotation. Following this, we analyse methods to automatize the manual process and analyse their performance in order to reduce the manual effort by providing an efficient method.

5.1 Resource Identification

In this section, we present the creation of a corpus for the gold corpus and how the property of Multimodality, Monolinguality, and Comparability are maintained. The need to create one arose when there were no such corpora publicly available to our knowledge. Even though creating a new corpus is only the first step towards our objective, it gives us the opportunity to study the corpus building process. Having this insight, exploring into the possibility of improving the alignment process would be a great motivation. As we know, majority of methods that solve NLP

tasks require some testing and possibly some training set corpus to evaluate their performance. As new NLP tasks arise, new aligned corpora would be required. This becomes a hurdle to the field of NLP due to the problem of scarceness of aligned corpora and more over the difficulty in creating one. Any improvement in the annotated corpus building process would directly help the field of NLP.

Corpus Properties

The corpus for our purpose should contain three property namely:

- Multimodality
- Monolinguality
- Comparability

Even though these properties have been discussed in Chapter 2, we revisit these properties and be more specific in terms of what exactly each property represents in the corpus we build.

Multimodality

The multimodality property indicates that the text corpus consists of information from different production modalities. We choose the production modalities of writing and speech where we deal with typed text and speech signals in the form of their manual transcription respectively. Even though it contains only texts, the corpus contains text that represent communication through these different information modalities.

Monolinguality

The monolingual property of a corpus is self explanatory indicating that the corpus consists of texts from a single language. These texts are all in the English language, especially American English, and are collected in a way that they all relate to a single topic and are of similar type. We select the topic on *Death of Diana* because there were sufficient amount of texts concerning this event that could be easily collected from different news sources of newswire and broadcast news. This relation of topic between the texts and the type of texts insures the property of comparability .

Comparability

Text Sources

The written texts were collected by selecting 12 newswire articles from a large collection of texts present in the North American News Text Corpus[1]. This corpus is a large collection of texts from different newswire sources and is distributed by the Linguistic Data Consortium (LDC). Among the 12 articles, 6 are from the Washington Post, 3 from Los Angeles Times, and 3 from New York Times. As mentioned earlier, these articles are based on the same topic and contain a total of 12,252 words. These information are summarized in table 5.1.

These articles are tagged using the Standard Generalized Markup Language, or SGML, annotations. The SGML tags are present to systematically include related information about the articles such as the id, author's name and even the structure of the article. A snippet of an article is shown in Figure 5.1. The figure shows the different tags which indicate certain information, i.e., the tag <*DOC*> indicates the beginning of the

[1]LDC catalogue no. :LDC95T21

Written Corpus				
Source	No. of Articles	Total Words	Total Segments	Sentences per Segment
Washington Post	6			
Los Angeles Times	3	12,252	291	1.8
New York Times	3			
Transcribed Corpus				
Source	No. of Articles	Total Words	Total Segments	Sentences per Segment
LDC Broadcast News	3			
ABC	1	18,604	403	2.7
CNN	3			

Table 5.1 – *Summary of the written and transcribed texts that constitute the MMC corpus*

article, <*TEXT*> indicates the beginning of the text, and <*p*> indicates the beginning of a paragraph.

```
<DOC>
<DOCID> latwp970902.0113 </DOCID>
<STORYID cat=i pri=r sel=tm--a> X3329 </STORYID>
<FORMAT> &D3; &D1; </FORMAT>
<SLUG> bc-diana-alcohol </SLUG>
<HEADER> 09-02 0564 </HEADER>
<PREAMBLE>
bc-diana-alcohol
 &UR; (wap) (ATTN: Foreign editors) &QL;
</PREAMBLE>
<BYLINE> By Rick Weiss </BYLINE>
<CPYRIGHT>
(c) 1997, The Washington Post
</CPYRIGHT>
<HEADLINE>
 &UR; Experts Say Factors Made Diana Crash Inevitable (Washn) &QL;
</HEADLINE>
<TEXT>
<p>
   WASHINGTON  &MD;  The blood alcohol concentration present in Princess
Diana's chauffeur increased his risk of being in a fatal, single
vehicle crash 300 to 600 times above normal, according to research on
the effects of alcohol on driving safety.
<p>
   Add to that the risk that comes with driving at high velocity  &MD;  the
vehicle is reported to have been hurtling at 120 mph, or the length of
two football fields every three seconds  &MD;  and of driving in darkness,
and it is almost inevitable that the car and its occupants would meet
a tragic end, experts said.
<p>
   ``A driver at that alcohol level and in those circumstances has
atrocious hand-eye coordination, delayed reaction responses, poor
decision-making, decreased vision and hearing,'' said Matthew Robb,
director of clinical services at Grace Clinic, which administers
Washington, D.C.'s educational program for people who have been
arrested for drinking while intoxicated.
<p>
   ``At 120 miles per hour, you need superhuman driving skills even in
the best of circumstances,'' Robb said. ``And this guy was not in the
best of circumstances.''
```

Figure 5.1 – *Snippet of an article which is present in the collection of the written texts*

To create a multimodal corpus, we collected transcripts of various audio/video speech on the same topic as the written texts. This was a challenge as there are very few transcripts on one particular topic. Most of the transcripts are directed towards building speech recognition systems

and were not meant to be of related topics. One such corpus is the *1997 English Broadcast News Transcripts*[2]. From this corpus, 3 transcripts were related to our topic and were selected. 4 other transcripts were manually transcribed from 50 minutes of speech from broadcast news from ABC and CNN. The transcripts from both of these sources consists of 18,604 words. These information are summarized in table 5.1. These transcript texts are tagged as the written texts as shown in Figure 5.2. The written and transcribed texts constitute the monolingual multimodal comparable corpus containing a total of 30,856 words. Once the texts for the corpus are collected, the next steps towards alignment is to define the text unit to align and the criteria for aligning them.

```
<episode filename="eo970830" program="CNN The World Today" language=English version=1 version_date=17-
Feb-98>

<section type=nontrans startTime=0.1 endTime=2.273>
</section>

<section type=report startTime=2.273 endTime=1614.183>
<turn speaker=Jeanne_Meserve spkrtype=female startTime=2.273 endTime=30.858>
<time sec=2.273>
 and ^Diana has been evacuated to a ^Paris hospital. {breath}
<time sec=5.583>
 %uh we know nothing about the nature of her injuries or the extent of her injuries at this time. But
again, ^Dodi ^Al ^Fayed, her companion, {breath}
<time sec=12.805>
 reportedly was killed in this crash in ^Paris tonight {breath}
<time sec=15.718>
 which happened about midnight ^Paris time. {breath}
<time sec=18.068>
 According to some reports they had had dinner at the ^Ritz Hotel {breath}
<time sec=21.065>
 and were on their way to ^Al ^Fayed's apartment.
<time sec=24.020>
 {breath} %uh ^Siobhan ^Darrow is %ah joining us from ^London. ^Siobhan,
<time sec=27.642>
 any reaction yet from authorities there in ^London?
</turn>
<turn speaker=Siobhan_Darrow spkrtype=female startTime=30.858 endTime=98.710>
<time sec=30.858>
 {breath} No. We have, we've been calling ^Buckingham Palace repeatedly. They're refusing to make any
official comment. {breath}
<time sec=37.244>
 All we have heard {breath}
<time sec=38.744>
 %ah is one, one comment from a spokesman that was reported in the wires earlier {breath}
<time sec=43.379>
 saying that this incident was an accident waiting to happen. He was referring to {breath}
<time sec=47.725>
 the fact that, %uh, there were apparently paparazzi chasing the car. {breath}
<time sec=52.256>
 %uh Princess ^Diana, as well as other members of the royal family, have repeatedly complained {breath}
```

Figure 5.2 – *Snippet of a transcript from the LDC98T28 corpus which is present in the collection of the transcript texts*

5.2 Segmentation

Before starting the annotation process for either the pair-wise or group-wise alignment, we have to define what is to be aligned. This defines the size of the text units to be aligned as mentioned in Section 2.2. One of the objectives of our annotation, presented in Chapter 1, is for it to be useful for information retrieval tasks where smooth navigation between text segments that contain specific information within different modalities is possible.

[2]LDC catalogue no. :LDC98T28

The optimal text segments for the purpose of searching specific information in some texts is to *define the text units such that each of them contains a single information*. In Section 4.1.1, we presented some of the state of the art methods to segment texts. It was mentioned that the comparisons between them would be difficult because the performance of each of those method varied while using different texts. On the other hand these text segmentation methods were not intended to find short text units containing a single information. Among these methods we tried C99 (Choi 2000) and Textiling (Hearst 1997) to segment our texts. The results of the two segmentation algorithm compared to the original text are given in tables 5.2, 5.3, and 5.4. The original text, as mentioned in section 5.1, is naturally segmented into 9 paragraphs and are delimited using the tag <*p*>. This same text was given to the MorphAdorner program [3] which segmented the text using Textiling and C99 algorithm. The result of Textiling was produced using the text window size of 10 tokens and for the C99 algorithm, the mask size is 11. Text Unit Criteria

From table 5.3 and 5.4 we see that Textiling and C99 algorithm produces fewer segments with larger text content which includes many information within them. Similar to these methods, other existing methods are not intended for the task of segmenting text according to information content but are rather targeted towards thematic or topical segmentation producing larger segments. Recent text segmentation systems as proposed by Guinaudeau et al. (2012) deal with large segments with more than 100 words per segment on average which is not adequate for short texts.

Our corpus contains two types of texts, the written texts and the transcribed texts. The natural structure of the texts in the corpus are different and will be treated slightly differently. The written text have a physical structure of paragraphs that are indicated in the files using the tag <*p*> as shown in Figure 5.1. These paragraphs are short with an average length of 1.8 sentences. In majority, the nature of these short paragraphs is that they usually contain specific information. Figures 2.2 and 5.1 demonstrate the preciseness and small size of paragraphs that may appear in the written news article text. For the written texts, these paragraph structures are selected as the text segments with a total of 291 text units. *Written text units*

There are 7 transcripts which come from two different sources: 3 transcripts were selected from the LDC corpus and the other 4 were manually transcribed. As the source of these transcripts are different, there are some differences in their physical structure. In Figure 5.2 we give examples of the transcripts. The transcript texts from the LDC corpus, as shown in Figure 5.2, have mainly two tags engulfing the text. One is the tag indicating the speaker of the text which is always one person, i.e. <*turn*>, and the second is the tag which indicates the time of the utterance of the text, i.e. <*time*>. There are no specified paragraph like structures as in the written text. The text units created by considering the time tag as the segment boundary are small pieces of text usually containing incomplete

[3] http://morphadorner.northwestern.edu/morphadorner/

1	WASHINGTON ; The blood alcohol concentration present in Princess Diana's chauffeur increased his risk of being in a fatal, single vehicle crash 300 to 600 times above normal, according to research on the effects of alcohol on driving safety.
2	Add to that the risk that comes with driving at high velocity the vehicle is reported to have been hurtling at 120 mph, or the length of two football fields every three seconds and of driving in darkness, and it is almost inevitable that the car and its occupants would meet a tragic end, experts said.
3	"A driver at that alcohol level and in those circumstances has atrocious hand-eye coordination, delayed reaction responses, poor decision-making, decreased vision and hearing," said Matthew Robb, director of clinical services at Grace Clinic, which administers Washington, D.C.'s educational program for people who have been arrested for drinking while intoxicated.
4	"At 120 miles per hour, you need superhuman driving skills even in the best of circumstances," Robb said. "And this guy was not in the best of circumstances."
5	Driving a car is a far more complex task than many everyday drivers appreciate, with constantly changing demands that require quick and often subtle responses. According to experts at the National Institute on Alcohol Abuse and Alcoholism, those skills can be divided into two categories: cognitive skills such as information processing, and psychomotor skills, which involve eye- rain-hand coordination. Both suffer badly under the influence of alcohol.
6	Princess Diana's chauffeur had undoubtedly lost significant control over his voluntary eye movements long before his blood alcohol concentration (BAC) reached the level of about 0.18 grams per deciliter detected at autopsy, according to NIAAA research. At BAC levels as low as 0.03, or about one-sixth those found in the chauffeur, the ability to focus briefly on an object as it passes by such as a concrete pillar and then refocus attention on the next, called "tracking," is seriously compromised. The driver's ability to steer precisely was also certainly undermined; studies have shown steering errors at blood alcohol levels as low as 0.035.
7	At levels of 0.04, less than one-fourth those found in Diana's driver, the attentional field begins to narrow. Lacking the bigger perspective, drivers are more likely to be surprised by an approaching object, and to overcompensate by pulling the steering wheel too hard.
8	The chauffeur had two things going for him that might have helped in less extreme circumstances: He was an experienced driver and a man. A 1989 study by the Arlington, Va.-based Insurance Institute for Highway Safety found that younger drivers and female drivers have a higher chance of being involved in fatal single vehicle accidents compared to more experienced drivers and male drivers with the same elevated alcohol levels.
9	But at the extremely high alcohol levels found in the chauffeur, no amount of experience could have helped, Robb said. "Judgment goes, so instead of calling off reckless activity you are much more prone to get swept up in the emotions of the moment," he said. "Years of training go by the wayside."

Table 5.2 – *Natural partition of a text in the written corpus indicated here by numbers which are indicated by the tag <p> in the original text. There are 9 text segments.*

Transcribed text units

information which do not satisfy the criteria of our short text. The text units, created by the turn tag as the segment boundary, may contain several sentences and may contain several information in them. But in many cases, the text units created by the turn tag gives an appropriate text unit with paragraph like structures, especially when there is some dialogue.

The manually transcribed files were intuitively segmented while transcribing. A segmentation boundary was inserted whenever the transcriber felt that a complete information has been spoken. The segmentation was done on the 3 transcripts selected from the LDC corpus. The turn tags were the basis of segmentation for these LDC transcripts which mostly produced small segments that were comparable to the paragraph segments in the written texts. There were some large text segments as well as

1	WASHINGTON; The blood alcohol concentration present in Princess Diana's chauffeur increased his risk of being in a fatal, single vehicle crash 300 to 600 times above normal, according to research on the effects of alcohol on driving safety. Add to that the risk that comes with driving at high velocity the vehicle is reported to have been hurtling at 120 mph, or the length of two football fields every three seconds and of driving in darkness, and it is almost inevitable that the car and its occupants would meet a tragic end, experts said.
2	" A driver at that alcohol level and in those circumstances has atrocious hand-eye coordination, delayed reaction responses, poor decision-making, decreased vision and hearing, " said Matthew Robb, director of clinical services at Grace Clinic, which administers Washington, D. C.'s educational program for people who have been arrested for drinking while intoxicated. " At 120 miles per hour, you need superhuman driving skills even in the best of circumstances, " Robb said. " And this guy was not in the best of circumstances. "
3	Driving a car is a far more complex task than many everyday drivers appreciate, with constantly changing demands that require quick and often subtle responses. According to experts at the National Institute on Alcohol Abuse and Alcoholism, those skills can be divided into two categories: cognitive skills such as information processing, and psychomotor skills, which involve eye-brain-hand coordination. Both suffer badly under the influence of alcohol. Princess Diana's chauffeur had undoubtedly lost significant control over his voluntary eye movements long before his blood alcohol concentration (BAC) reached the level of about 0.18 grams per deciliter detected at autopsy, according to NIAAA research. At BAC levels as low as 0.03, or about one-sixth those found in the chauffeur, the ability to focus briefly on an object as it passes by such as a concrete pillar and then refocus attention on the next, called " tracking, " is seriously compromised. The driver's ability to steer precisely was also certainly undermined; studies have shown steering errors at blood alcohol levels as low as 0.035. At levels of 0.04, less than one-fourth those found in Diana's driver, the attentional field begins to narrow. Lacking the bigger perspective, drivers are more likely to be surprised by an approaching object, and to overcompensate by pulling the steering wheel too hard. The chauffeur had two things going for him that might have helped in less extreme circumstances: He was an experienced driver and a man. A 1989 study by the Arlington, Va. -based Insurance Institute for Highway Safety found that younger drivers and female drivers have a higher chance of being involved in fatal single vehicle accidents compared to more experienced drivers and male drivers with the same elevated alcohol levels. But at the extremely high alcohol levels found in the chauffeur, no amount of experience could have helped, Robb said. " Judgment goes, so instead of calling off reckless activity you are much more prone to get swept up in the emotions of the moment, " he said. " Years of training go by the wayside. "

Table 5.3 – *The text in 5.2 segmented using Textiling Algorithm with a sliding window size of 10. 3 text segments were produced and are indicated using numbers.*

in the LDC segments, but in fewer number and were manually segmented making the segmentation process fast and achievable.

Initially, there were 308 text segments generated from the manual segmentation of the manually transcribed text along with the LDC texts segmented on the basis of the turn tag. After further segmentation of the transcripts, the texts had 403 segments. With these written and transcribed text segments, our MMC corpus consists of a total of 694 text segments. Table 5.1 gives the summary of the properties of this corpus.

5.3 Alignment Criteria

In addition to the text segments, we need a criteria to obtain what we believe is a good alignment. Our criteria for alignment is similarity between the text segments, or in other words if two text segments are similar they

1	WASHINGTON ; The blood alcohol concentration present in Princess Diana's chauffeur increased his risk of being in a fatal, single vehicle crash 300 to 600 times above normal, according to research on the effects of alcohol on driving safety. Add to that the risk that comes with driving at high velocity the vehicle is reported to have been hurtling at 120 mph, or the length of two football fields every three seconds and of driving in darkness, and it is almost inevitable that the car and its occupants would meet a tragic end, experts said. ' ' A driver at that alcohol level and in those circumstances has atrocious hand-eye coordination, delayed reaction responses, poor decision-making, decreased vision and hearing, " said Matthew Robb, director of clinical services at Grace Clinic, which administers Washington, D. C.'s educational program for people who have been arrested for drinking while intoxicated. ' ' At 120 miles per hour, you need superhuman driving skills even in the best of circumstances, " Robb said. ' ' And this guy was not in the best of circumstances. " Driving a car is a far more complex task than many everyday drivers appreciate, with constantly changing demands that require quick and often subtle responses.
2	According to experts at the National Institute on Alcohol Abuse and Alcoholism, those skills can be divided into two categories: cognitive skills such as information processing, and psychomotor skills, which involve eye-brain-hand coordination. Both suffer badly under the influence of alcohol.
3	Princess Diana's chauffeur had undoubtedly lost significant control over his voluntary eye movements long before his blood alcohol concentration (BAC) reached the level of about 0.18 grams per deciliter detected at autopsy, according to NIAAA research. At BAC levels as low as 0.03, or about one-sixth those found in the chauffeur, the ability to focus briefly on an object as it passes by such as a concrete pillar and then refocus attention on the next, called ' ' tracking, " is seriously compromised. The driver's ability to steer precisely was also certainly undermined; studies have shown steering errors at blood alcohol levels as low as 0.035. At levels of 0.04, less than one-fourth those found in Diana's driver, the attentional field begins to narrow. Lacking the bigger perspective, drivers are more likely to be surprised by an approaching object, and to overcompensate by pulling the steering wheel too hard. The chauffeur had two things going for him that might have helped in less extreme circumstances: He was an experienced driver and a man. A 1989 study by the Arlington, Va. -based Insurance Institute for Highway Safety found that younger drivers and female drivers have a higher chance of being involved in fatal single vehicle accidents compared to more experienced drivers and male drivers with the same elevated alcohol levels. But at the extremely high alcohol levels found in the chauffeur, no amount of experience could have helped, Robb said.
4	' ' Judgment goes, so instead of calling off reckless activity you are much more prone to get swept up in the emotions of the moment, " he said. ' ' Years of training go by the wayside. "

Table 5.4 – *The text in 5.2 segmented using C99 Algorithm with a mask size of 20. It produced 4 text segments and are indicated using numbers.*

are aligned. We follow our intuitive similarity definition presented in Section 2.2.2. It states that :

Two text segments are similar if they contain at least one common main information.

Here is an example of a short text pair that are similar according to the alignment criteria :

Text Segment I : Two prideful patrician families agreed Monday with the British government to give Princess Diana a "unique funeral" that will combine the families' wishes for privacy with public demands to honor a woman who touched the nation's heart.

Text Segment II : Officials said Diana's family and members of the royal family concluded that Diana should have "a funeral that reflected her life," a palace spokesman said. "A unique funeral is being devised."

This criteria of alignment, performs a fine-grain analysis of the text segments on the basis of information. In the example above, the only main information in the two short texts is *Diana to receive a unique funeral* which is common between them satisfying the criteria for alignment. Unlike these two texts, there are several short texts that contain more than one information which may increase the difficulty in analysing them. The two main problems that may arise are: the identification of the main informations and finding the commonality between them. For example, in the Text Segment III below, the identification of the main information is not straight forward.

Text Segment III : Reports say that motorcycle-riding paparazzi, that voracious breed of celebrity photographers, were in hot pursuit of Princess Diana and her companions when the princess' car smashed into the side of an underpass in Paris, killing her and two other passengers.

Some may consider Text Segment III to contain two different main informations, first is the *paparazzi were in hot pursuit of Diana* and the second is the *car crashed killing three passengers* which substantiates the previous information, whereas some may consider it to contain only one main information which is *paparazzi were in hot pursuit of Diana* and the other information, *car crashed killing three passengers*, to be just an information as it is not the central point. The consequence of the problem of identifying the main information is carried on to the subsequent step of finding the common information between them making it problematic. In addition to this, the identification of commonality in itself is tricky. Below is an example of such a pair:

Segment IV : News agencies quoted prosecutors as saying Paul's blood level was 1.75 grams of alcohol per liter of blood, while the legal limit for driving in France is 0.5 grams the level after about two glasses of wine. The 0.5-gram limit is equivalent to a blood alcohol content of .06 percent, making the French law slightly stricter than those of most U.S. states, which set the limit at .08 or .10 percent blood alcohol content.

Text Segment V : The announcement did not specify his blood alcohol level, but news agencies quoted official sources as saying it was three times the legal limit for drivers in France.

In the text segments IV and V, the main information about the blood level being high is not apparent. The segments have to be read carefully to first determine the main information and then comparing them to see if they do convey the same message. This takes some time and analysis which makes the comparison difficult. One source of these problem is that the criteria of alignment does not give a definition for the main information. But as you can see from these examples given above, specifying

a definition of the main information is hard to accomplish as mentioned by Hatzivassiloglou et al. (1999). This makes our definition for similarity intuitive and subjective in nature. In addition to this nature, the problems that come with the criteria may tempt us to annotate the text pairs with the degree or scale of fulfilment of the criteria indicating how similar the texts are with respect to our definition.

In terms of property, as stated in Section 2.2.2, the similarity definition is bi-directional, i.e. if $A \sim B$ then $B \sim A$, but does not necessarily satisfy the property of transitivity, i.e. if $A \sim B$ and $B \sim C$, then not necessary that $A \sim C$. With these understanding of the alignment criteria and the text segments, we are able to start the alignment process.

5.4 Pair-wise Manual alignment

Our corpus consists of a total of 694 text segments from two different modalities which gives a total number of 240,471 possible text pairs from equation 2.1. Among these text pairs, we try to select the pairs that satisfy the similarity definition for alignment. Due to this huge amount of text pairs, we follow the divide and conquer paradigm by adapting the following alignment step:

- we first perform the alignment inside each modality
- then perform the alignment across modalities.

In our case, this implies that the alignment is carried out three times as shown below:

- Alignment between texts in the written part
- Alignment between texts in the oral part
- Alignment between written and oral texts of the multimodal corpus

But to align these texts, we do not already have an automatic method. Our alignment problem is unique and there is no automatic method, as far as we know, that can align these short texts automatically as our objective is to find a method that does so. One possible way of getting this corpus aligned is to do it manually. We start the alignment process in the traditional pair-wise manual alignment method as presented in Section 2.2.

The first task is to manually align the written text of the MMC corpus. This part consists of 291 segments. Among these, the segments that contain more than 10 words were selected as the segments to be aligned. This was done to filter out the noise such as sentences without any information relating to the topic which was present in the text due to the creation of text segments on the basis of tags. This filtering gave 240 text segments which gives a total of 28,680 text pairs to compare. This is still large to compare manually because going through them one at a time and deciding whether they are similar or not would take a large amount of time and

effort. This motivated us to reduce the search space in which similar text pairs would be selected. To solve the problem of this large search space, the alignment problem is further broken in two different phases. The first phase reduces the number of alignment pairs from its original pairs and in the second phase, the aligned pairs are extracted from the reduced search space.

5.4.1 First Phase

First phase of manual alignment

The first phase of manual alignment is the part in which the total combination of alignment pairs are reduced by selecting candidate alignment pairs. This is done such that the actual alignment pairs is in the subset of the candidate alignment pairs.

$$ActualAlignmentPairs \subset CandidateAlignmentPairs \tag{5.1}$$

With these candidate alignments, the annotators have a smaller set of text pairs to work with. By concentrating on a smaller set of alignments, we increase the effectiveness of the manual alignment. To reduce the set of alignments we applied the alignment criteria which is presented below :

Criteria for Candidate Pairs

A text pair is selected as a candidate pair if they share at least one concept.

The concepts that we focus on are of Noun phrases and Verb phrases. The information contained in a phrase can be expressed with different surface forms. For instance, in the Text Segment VI and VII below, the phrases *car* and *Mercedes S-280* represent the same concept but are different surface forms.

Text Segment VI : Diana and Fayed were trying to escape motorcycle-borne scandal-sheet photographers, the tabloid paparazzi who have hounded the princess ever since her engagement to Prince Charles in 1980, when the **car** spun into the wall at an estimated 100 miles per hour.

Text Segment VII : Widespread reports here have said the sedan, a **Mercedes S-280**, was traveling at a speed of at least 90 mph in a 30-mph zone when it struck a concrete pillar in the tunnel under the Alma bridge in central Paris, and possibly much faster. Reports Monday said the car's speedometer was frozen at 196 kilometers per hour, or 121 mph.

These two text segments are examples of candidate pairs. They have couple of noun and verb phrases that share the same concept. In Table 5.5 we give the phrases that have the same concept in the text pair. We would prefer these type of pairs to be collected for further analysis in the second phrase. Even though these text segments are not similar on the basis of our similarity definition, they satisfy the criteria for candidate alignments and will be collected as a candidate alignment. By using the surface form we would have missed this common concept. This criteria to select candidate alignment will therefore theoretically guarantee that the actual aligned text segment pairs will be present in the list of candidate

alignments. And because the overlapping concepts are handled properly by humans they are easy to spot, the candidate pairs can be efficiently selected. In the two texts above, there is no overlap of words other than stop words. This is one of the reasons why simple automatic methods such as the overlap of words or even cosine similarity would not be helpful for the extraction of the candidate alignments.

Phrase I	Phrase II
car	Mercedes S-280
struck a concrete pillar	spun into the wall
high speed	100 miles per hour

Table 5.5 – *The same concept expressed using the different surface forms.*

5.4.2 Second Phase

Second Phase

Once these candidate alignments are collected, they need to be examine to decide on their similarity. To examine the candidate alignments, they are given to the annotators for annotating the actual alignments which is the second phase. The number of candidate alignments will be less than the original combination of pair of text segments and therefore many annotators can work efficiently on the small set in less human hours. The actual alignments are selected by the annotators using the selection criteria of similarity defined in Section 5.3.

As mentioned in section 5.3, there could be two possible ways to annotate the text pairs. The first is to make a binary decision on whether a text pair satisfies the similarity criteria or not. This way of annotating would be an ideal scenario of annotation. But the expressiveness of natural language is so diverse that this binary way of annotating may not be practical as can be seen from the text segment examples provided in section 5.3. In these examples, especially text segments IV and V provide evidence that the binary decision might not be enough to express similarity with the similarity criteria we define.

Natural language is able to present an information in different ways for instance by giving specific information, elaborating that information, or providing implications of such information. The different presentation of the same information could make the identification and commonalities of the main information sometimes tricky. One solution would be the second way of annotating text segments where we express the degree of fulfilment of the similarity criteria by a numerical scale or degree of fulfilment.

Using a numerical scale, for instance from 0 to 5, to indicate fulfilment of the similarity criteria could be a solution where 0 indicating non and 5 indicating complete fulfilment. But this scale would be difficult to interpret as the scale is not restrictive enough for the proper interpretation or documentation of the fulfilment. Another solution is to use the degree

or categories of fulfilment, which is similar to the numerical scale but becomes more restrictive in nature as the categories specifies some guidelines. Even though the underlining selection of similar pair is based on the similarity criteria, the categories would help perceive the true nature of our similarity definition and the varieties of pairs selected by the criteria. We create 3 categories of fulfilment of the similarity definition. These categories are *exactly similar*, *similar*, and *nearly similar* . Examples of these different types of category are shown in the Table 5.6.

Categories of Aligned Pairs

Exactly Similar Pair
On a sun-dappled day that tasted of approaching autumn, patient mourners waited as long as five hours to sign condolence books at St. James Palace, where Diana rests in a closed coffin before the altar in the 450-year-old Chapel Royal.
Thousands of people lined up to sign official books of condolences at St. James' Palace, where Diana's coffin will lie in private until the funeral, with mourners waiting up to six hours to make their way into the room in the palace where the books are on display.
Similar Pair
A volunteer firefighter was working on Trevor Rees-Jones, the bodyguard in the front passenger seat who was to be the only survivor. Mailliez confirmed Tuesday that Rees-Jones was the only passenger in the Mercedes who was wearing a seat belt.
A fourth occupant of the Mercedes, bodyguard Trevor Rees-Jones, was the only survivor of the accident, and the only person in the car believed to have been wearing a seat belt. Rees-Jones was sitting in the front passenger's seat, and suffered lung, head and facial wounds.
Nearly Similar Pair
Diana and Fayed were trying to escape motorcycle-borne scandal-sheet photographers, the tabloid paparazzi who have hounded the princess ever since her engagement to Prince Charles in 1980, when the Mercedes spun into the wall at an estimated 100 miles per hour.
French television said Diana was being pursued by paparazzi when the crash occurred, and French Interior Minister Jean-Pierre Chevenement said police were questioning seven photographers as part of a criminal investigation into the accident.

Table 5.6 – *Examples of the three different categories of similar text in the actual alignments selected by the annotator.*

In the *exactly similar* category, all the main information in one text segment is present in the other segment and vice versa. For instance, in our example of the similar category there are two main information. The information about *mourners waited to sign condolence book at St. James Palace* and the *Diana's coffin is in St. James Palace* .

In the *similar* category, both the text segments have at least one main information in common which can be seen in its example. The information about *Ree-Jones was sitting in the passenger's seat who was the only one wearing a seat belt and the only one who survived* is common but one text segment has another main information about *Ree-Jones suffering lung, head, and facial wounds* which is not in the other text segment.

The third category of *nearly similar* pairs have at least one main information in common but are presented in different form such as an

implication of the information as shown in the example. Both of the text segments convey the main information about *Diana was being followed by paparazzi while crashed* but presented differently. The first text mentions that Diana was escaping the scandal-sheet photographers and spun into the wall which helps understand using the context that the photographers who were following them were paparazzi and the phrase spun into the wall indicates they crashed. In the second text, the information is clearly given as Diana was being pursued by paparazzi when the crash occurred. There are other information beside this, for instance the information about the car crashed at the speed of 100 miles per hour but this information could be considered as it is not the main information. This is a problem with the intuitive definition of the similarity definition explained earlier in section 5.3, it gives the freedom to the annotator to decide which information are the main ones.

We annotate the candidate alignments extracted from the written corpus part with both these two ways of second phase alignment. The result of these manual annotation processes on the written part of the MMC corpus are presented in the next section.

5.4.3 Results

The written text portion of the MMC corpus consists of a total of 28,680 text pairs. The first phase of the annotation process was carried out by a single annotator who selected 3,418 candidate alignment pairs from the total text pairs by following the similarity criteria. As this first phase is done manually, this gives the freedom to the annotator to remove any selected candidate alignments if an easy decision can be taken that the alignment is not of any use without any doubt to further reduce the search space. This selection process took about 71 hours.

In the second phase, the selection of actual alignments were selected from the candidate pairs. This alignment was done twice, once with the binary annotation where we decide on whether the similarity criteria was satisfied or not and the second time we annotate using the degree of fulfilment of the similarity definition. The first run of annotation using the binary annotation was carried out by two annotators that were given the candidate alignments to select the actual alignments. The selection process was carried out independent to each other. The annotator agreement between them was measured using the Kappa statistics, k, and k=0.5, which indicates the agreement between them to be moderate (Artstein and Poesio 2008). The error that is present between the annotators is most likely due to our intuitive definition of similarity. The disagreement between the annotators were resolved by reasoning between the two annotators.

The two annotators took about 20 hours on average to complete this phase. This indicates that to analyse each pair, it takes about 21 seconds. In this phase, they selected 144 actually aligned text pairs out of the 3,418 candidate alignments from the first phase. The total time for annotating

the written text pairs which includes the two phases is about 91 hours. If 28,680 initial pairs were compared to select the actual pairs directly, the total time would take about 166 hours assuming that it would take 21 seconds for analysing each pairs. Our two phase method saved 75 hours of human effort to annotate the total written text pairs. Having said that, there was only 144 pairs that were the actual alignment out of the 28,680 initial total pairs. The amount of actually aligned pairs are only 0.5% of the total pair. The summary of the alignment process is given in table 5.7.

Alignment Phase	Initial text pairs	Candidate pairs	Actual pairs	Time (hrs)
1^{st}	28,680	3,418	-	71
2^{nd}	3,418	-	144	20
Total		3,418	144	91

Table 5.7 – *Summary of the text pairs generated from the alignment process of the written text portion of the corpus using the binary annotation.*

In the second run, the two annotators were given the set of candidate pairs to extract the actual aligned pairs on the basis of the degree or category of fulfilment of the alignment criteria. These categories are *exactly similar*, *similar*, and *nearly similar*. As in the previous annotation, the annotations were carried out independently of the annotators and the disagreements were resolved by reasoning. The agreement between the annotator has a kappa of 0.41 indicating the agreement is moderate. The total number of aligned pairs extracted were 418 which is about 3 times the number extracted using the binary annotation. Among these actual pairs there were 5 exactly similar pairs, 142 similar pairs and 271 nearly similar pairs.

One of the reasons behind this big difference between the alignments extracted by the binary annotation and by the category annotation is that the binary decision requires a strict way of deciding whether a pair is similar or not, and this decision is usually affected by previous decisions because the identification of the main information is intuitive. While with the categorical annotation, decisions are more flexible and the decision of one pair being similar or not solely depends on that pair alone. So, it doesn't require any reference of other pairs, but on the other hand requires more time on deciding its category. Beside the ease of comparison and flexibility, categorical based alignment provides help to the annotator by giving a direction to identify information within the text. The 144 text pairs extracted using the binary annotations are all present in the 418 pairs extracted by the category annotation. The summary of this annotation is given in table 5.8.

The tables 5.7 and 5.8 shows that most of the effort used to find the actual alignments are wasted on analysing pairs that are not similar. So even though the two phase method of manual annotation reduces the time of annotating, this method is still time consuming and difficult as manual effort has to be done. Using this manual method to annotate the rest of the text in the corpus would not be a practical solution which calls for some

Alignment phase	Initial pairs	Candidate pairs	Actual pairs	Exactly Similar pairs	Similar pairs	Nearly Similar pairs
1^{st}	28,680	3,418	-	-	-	-
2^{nd}	3,418	-	418	5	142	271

Table 5.8 – *Summary of the alignment process of the written text based on the category based fulfilment of the alignment criteria.*

automatic method to be used in the selection of the candidate alignments. In the next section, we describe a new text representation method that is capable of representing texts and in turn could be used to find candidate texts. This method can be used for the automatic selection of candidate alignments which will further reduce the time and effort for annotating the rest of the texts in the corpus.

5.5 Pair-wise Hybrid Alignment

As shown in the previous section, the two phase manual alignment saves time in annotation but still a lot of manual effort is devoted to extract few actual alignments. If we could at least automatically extract the candidate alignments automatically, then the amount of manual effort in annotating the corpus will reduce. It took approximately 71 hours of manual effort to extract the candidate alignment in the first phase. If this could be done automatically, then we would reduce this annotation process to 20 hours rather than 91 hours. This would make the annotation of the rest of the MMC corpus efficient and fast.

5.5.1 Experiments

In Section 3.1 and 3.2, different text representation methods and similarity measures were presented along with their properties. These methods are used to find similar texts and could also be used for the automatic extraction of candidate alignments. The vector representation of texts has a main role to play while finding similar pairs because it represents the text from which different metric computes a similarity value. The most simple of representation is the Vector Space Model (VSM) which represents texts as a matrix of vectors with respect to terms but with an assumption stating that, more the overlap of important terms between texts the more similar they are (Salton et al. 1975). Another state of the art representation of text is LSA which is generated by using the Singular Value Decomposition (SVD) method to perform mathematical transformation of the matrix provided by VSM. More details on these methods are given in chapter 3.

We use these two methods to analyse how they perform in the task of candidate alignment extraction. The experiments were performed on the annotated written texts of the corpus that were manually aligned based on the binary annotation and multi-category annotation as explained in section 5.4.2. As this corpus is manually aligned, this will be a good set to test the different methods. Some properties of this written texts are

presented in table 5.1.

Properties of automatic method

A good method to extract candidate alignments should be able to retain almost all the actual alignment pairs and reduce the initial set of pairs to a minimum. In other words, the automatic method should be able to reduce the initial text pairs as low as possible while maintaining a high recall of actual alignments. Another important property the method should have is the slow rate of decrease in recall with respect to the thresholds of the method which makes sure that with small variations in the threshold the recall will not have a drastic change. The decrease rate is an important property of a method to select candidate alignments because we require to select a threshold which determine how many candidate pairs are selected. This threshold will not only be for the written text but for all possible texts. Due to the different nature of texts, choosing a method with the least decrease rate in the recall will prevent a large loss if the threshold is not suitable for some particular text. This will allow us to use the best method with the most safe threshold value to find candidate alignments in other modalities without the fear of losing many actual alignments.With these properties we could compare the two methods.

Among the various similarity metric, we use Cosine, Euclidean, and Jaccard similarity measures to find a similarity value from the text representation given by Vector Space Model (VSM) and Latent Semantic Analysis (LSA). Using the manual annotations we calculate the number of candidate alignments extracted by each method. The local weights and the global weights whose product is used for the standard weights of a term are explained in section 3.1.1 and are again presented in Table 5.9 and 5.10. With three local weights and the possible product of different combination between the local weights and the global weights, 15 possible weights can be formed. All these weights are used with VSM and LSA to see how they effect the extraction methodology.

Type	Local weights
tf	$L(i,j) = tf(i,j)$
logtf	$L(i,j) = log(tf(i,j) + 1)$
bintf	$L(i,j) = bin(i,j)$

Table 5.9 – *The local weights of a term i in document j in terms of the term frequency tf, the logarithm of the term frequency logtf and the binary term frequency bintf.*

Type	Global weights
norm	$G(i) = 1/sqrt(\sum_j L(i,j)^2)$
gwidf	$G(i) = gf(i)/df(i)$
idf	$G(i) = 1 + \log(\text{ ndocs } /df(i))$
ent	$G(i) = 1 + \sum_j p(i,j) \log p(i,j) / \log ndocs$

Table 5.10 – *The global weights of the term i in document j in terms of the global frequency gf, normalized value norm, inverse document frequency idf and the global entropy ent where ndocs is the total number of documents, df is the term frequency in a document and $p(i,j) = tf(i,j)/gf(i)$*

For the experiment, the function words in the text are removed from the written texts and the remaining words were stemmed using Snowball [4] which uses Porter stemmer. We evaluate the performance of the methods using the number of text pairs retrieved at different thresholds and the average recall decrease rate (RDR). This is the percentage of decrease in the recall from one threshold to the next and is calculated using the first four changing recall values. For instance, if the recall at threshold A is R_A and recall at threshold B is R_B, the recall decrease rate between these two thresholds is given by :

$$\frac{R_A - R_B}{R_A} \times 100 \tag{5.2}$$

The best results with respect to the average RDR produced by different combination of similarity measures along with the text representation methods are presented in table 5.11 and in table 5.12, we show the change in recall value corresponding to each similar pair category at different threshold.

Method	LSA		VSM	
Metric	Cosine		Cosine	
Weight	tf * entropy		tf * entropy	
Threshold	Retrieved	Recall	Retrieved	Recall
0.0	21485	0.99	28680	1
0.1	11851	0.96	8222	0.93
0.2	6310	0.88	2617	0.76
0.3	3135	0.75	730	0.47
0.4	1308	0.56	191	0.21
0.5	479	0.31	42	0.06
0.6	157	0.15	7	0.01
0.7	49	0.06	0	0
0.8	11	0.01	0	0
0.9	0	0	0	0
Average recall decrease rate	8.13		17.38	

Table 5.11 – *The VSM and LSA representation with the similarity measure and weights which produces the least average decrease rate in recall compared with different combinations of weights and similarity metric on the written part of the corpus annotated using the binary type annotation.*

In table 5.11, we can see that VSM has a big average recall decrease rate. This indicates that VSM is highly discriminative while extracting candidate pairs which we want to avoid. LSA on the other hand has comparatively low average RDR indicating LSA has a smooth transition of extraction of pairs which ensures it to be more robust against error that may occur while selecting the threshold. This comparison also holds on the multi-category annotation of the written texts as shown in Table 5.12. For all the different types of similarity category, LSA gives the least average RDR value. Even though the average RDR for LSA is lower than VSM,

[4] www.http://snowball.tartarus.org/

th	LSA			VSM		
	es	s	ns	es	s	ns
0.0	1	1	0.99	1	1	1
0.1	1	0.96	0.96	1	0.92	0.93
0.2	1	0.88	0.88	1	0.77	0.76
0.3	1	0.74	0.75	1	0.51	0.45
0.4	1	0.59	0.54	0.80	0.24	0.18
0.5	0.8	0.36	0.28	0.40	0.08	0.04
0.6	0.4	0.17	0.13	0.20	0.02	0.0
0.7	0.4	0.05	0.05	0.0	0.0	0.0
0.8	0.4	0.01	0.01	0.0	0.0	0.0
0.9	0.0	0.0	0.0	0.0	0.0	0.0
Avg.RDR	20	8.67	8	26.7	16.33	18.33

Table 5.12 – *The different recall values at different threshold value, th, and the average recall decrease rate (RDR) for exactly similar, es, similar, s, and nearly similar, ns, using LSA and VSM methods on the written part of the corpus annotated with the multi-category type annotation.*

LSA do not give good results compared to the number of retrieved candidate alignments. Looking at the number of retrieved pairs in table 5.11 in the range of 0.9 recall, the least number of retrieved pairs extracted is 11,851. This number of retrieved pairs is very different from the manually selected candidate pairs. Due to these short comings of the state of the art methods, we propose a new method called Short text Vector Space Model which is explained in the next section.

5.5.2 Short text Vector Space Model (SVSM)

In the first phase of the manual annotation process the candidate alignments were selected on the basis of the overlap of concepts of elements and to detect this overlap we would require more than word overlap. There are other information within a text other than term and their weights that has to be incorporated to represent a text to its fullest. For instance, in linguistics the notion of text cohesion exist which is defined as, *"A property of text whereby certain grammatical or lexical features of the sentences of the text connect them to other sentences in the text"* (Hoey 1991). Halliday and Hasan (1976b) define cohesion as a network of relationships between locations in the text, which could be both intrasentential and intersentential. They used *lexical cohesion*, which is the reiteration and collocation of terms, to find similarity between sentences which in turn helped them in segmenting text. As the collocation and reiteration can be calculated within a text segment, e.g. documents or window of words, the lexical cohesion property within our corpus could be exploited to find similar segments.

Text Cohesion

Each term has a set of context, or "profile" as mentioned in Kaufmann (2000), within which terms occur together regularly to have a relationship of collocation creating strong bonds between them. This relation has been

stressed by Halliday and Hasan (1976b) :

Without our being aware of it, each occurrence of a lexical item carries with it its own textual history, a particular collocational environment that has been built up in the course of the creation of the text and that will provide the context within which the item will be incarnated on this particular occasion

Non of the semantic spaces presented in Section 3.1 explicitly incorporates these text properties. Kaufmann (2000) tried to incorporate the lexical cohesion, called second order cohesion, to find similarity between windows of words by representing text using the VSM model where term vectors have dimensions representing content words. The content words are a fixed number of meaningful words present in the domain dictionary. The weights for each term corresponding to the content word is the co-occurrence between them within a fix window. A text is represented by the summation of these term vectors. In this model, the idea of summing the term vectors to represent the text is a very useful step. With this summation, the resulting text vector will inherently incorporate information of co-occurrence and context overlap of terms which will represent the text more explicitly which will eventually help in the similarity process. One problem in this method is the use of the content or meaningful words and in addition to this the collocation of terms is calculated only with the content words which might not give a complete representation. This text representation is then used to find the similarity between windows of consecutive words in order to find text segments which performed better than the TextTiling method (Hearst 1997) which uses text overlap to find how similar two texts are.

Short text Vector Space Model

The new text representation that we propose is called Short text Vector Space Model (SVSM) which aims at capturing the reiteration and collocation property of terms to represent the text. This will help in finding similar text which is accomplished by using the distribution of the terms as well as the collocation information with the summation of term vectors as done by Kaufmann (2000). SVSM starts by transforming the text representation produced by the VSM model where texts are represented as term-text matrix with weights. The weights could be any weight function but for the purpose of explaining we take idf value of the term as weights. Some of these functions are presented in section 3.1.1. This vector space does not only give the importance of each term for some text but also the distribution of the term throughout every text. This information is used by SVSM to get the reiteration and the collocation information. For each text, a text vector is created from the term vectors. Given a corpus C of m texts and n terms, the term vector, $\vec{t_j}$, for term t_j is a vector created with m number of possible dimensions where each dimension represents a unique text. The presence of the term in a text is indicated by its text id, P_i and the term's inverse document frequency, idf, here each text is considered as a document, as shown below:

$$\vec{t_j} = [(P_1, idf_j), (P_5, idf_j), ..., (P_i, idf_j)]$$

here, P_i is included to easily indicate the corresponding text where the

term t_j is present, $i \in 1, .., m$ and idf_j is the idf value of term t_j. This term vector is a reduced vector space representation where text that do not contain the term is absent which saves space. The dimension of the matrix formed by term vectors can be further reduced using Latent Semantic Analysis (Deerwester et al. 1990) or Principle Component Analysis (Jolliffe 1986).

Once we have the term vectors, we can create text vectors by the vector summation of the term vectors for the terms present in that text. For instance, to create a text vector for a text consisting of terms $t_1, t_2, .., t_k$, the resulting summation of the vectors will have the same number of dimensions as in the term vector where, each dimension corresponds to a text. The dimension d_i, of the text vector corresponding to the text P_i will be $d_i = \Sigma_{j=1;t_j \in P_i}^{k} idf_j$, where idf_j is the idf value of the term j and $i \in 1, .., m$. This process of text representation is simplified in figure **??**.

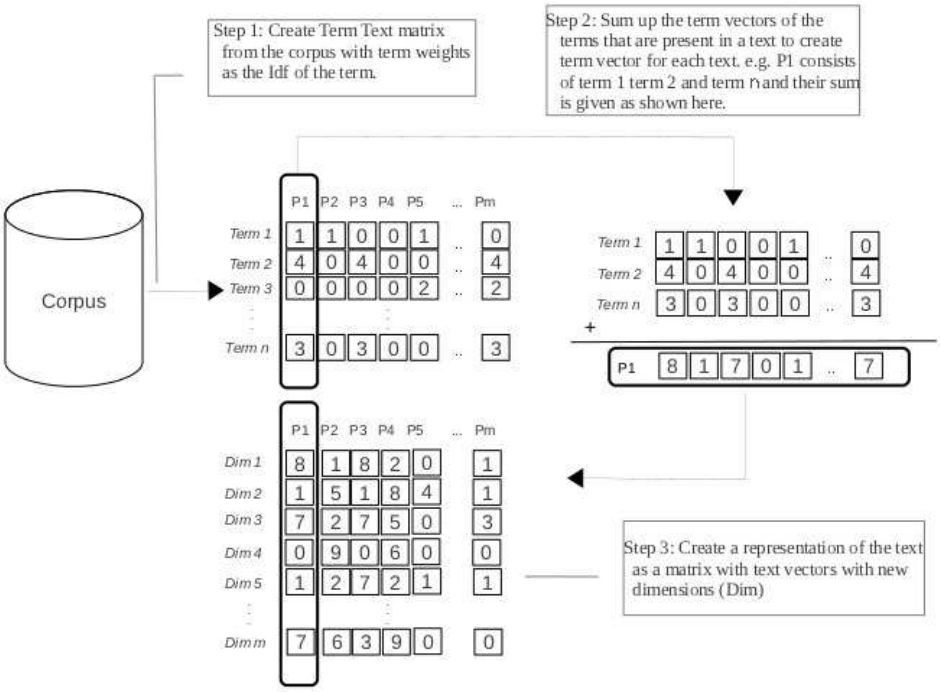


Figure 5.3 – *The steps for representing text using SVSM.*

The new representation of the corpus is a square matrix with the same number of rows and columns which equals to the number of text present in the corpus. Each dimension across the column of the matrix, indicated by P_i, represents a text as it is the sum of the idf values of the terms that are present in that text. This dimension now acts as a template for the text that it represents. Each text expresses some information using the terms that it contains and these terms are distributed through out the corpus trying to express a set of information giving us the distribution and a set of information. As each information is connected to some topic or subtopic, the distribution of terms indicate the distribution of information which becomes vital in representing the text. The combination of term

vector will give information about terms being together throughout the corpus capturing the collocation and reiteration information. The summation of idf gives a measure of the proportion of important terms that are present. The idea behind this representation is that the distribution of similar important terms or co-occurring terms indicates the closeness of the text.

This model like the VSM model makes an assumption that the dimensions of the text vector are independent to each other but in practice we know that texts in corpus are somehow related. Even though this assumption may produce incompleteness issues, the inclusion of lexical cohesion makes the model closely represent the text. SVSM produces denser matrix than the VSM model. In VSM, representation of short text creates a sparse matrix because of the fact that in short texts the repetition of terms is reduced compared to documents and hence VSM loses its power of representation as it mainly focuses on term repetition.

Performance of SVSM on Written texts

The comparison between these methods will be clearer once we use them to select the candidate alignments . We compare the performance of SVSM in extracting the candidate pairs with VSM and LSA. The results of SVSM are placed along VSM and LSA in table 5.13.

Best Method	SVSM		LSA		VSM	
Metric	Cosine		Cosine		Cosine	
Weight	tf		tf * entropy		tf * entropy	
Threshold	Retrieved	Recall	Retrieved	Recall	Retrieved	Recall
0.0	28680	1	21485	0.99	28680	1
0.1	25777	1	11851	0.96	8222	0.93
0.2	18718	0.99	6310	0.88	2617	0.76
0.3	13160	0.98	3135	0.75	730	0.47
0.4	8546	0.91	1308	0.56	191	0.21
0.5	4420	0.82	479	0.31	42	0.06
0.6	1779	0.64	157	0.15	7	0.01
0.7	498	0.35	49	0.06	0	0
0.8	66	0.09	11	0.01	0	0
0.9	0	0	0	0	0	0
Average recall decrease rate	2.87		8.13		17.38	

Table 5.13 – *The SVSM representation, along with LSA and VSM, with the weight and similarity metric that produces the least average decrease rate in recall with three different similarity metric on the written part of the corpus annotated using the binary type annotation.*

This table shows that SVSM performs the best in terms of the average RDR which uses the cosine similarity metric and tf weights. It gives a good output at the threshold of 0.4. The recall is 91 percent meaning out of 418 actual pairs about 380 actual pairs were present with 38 actual pairs being missed, which includes mostly nearly similar pairs, while selecting only 8,546 candidate alignments. This number of candidate alignment is better than the one produced by LSA with the recall value in the range of

th	SVSM			LSA			VSM		
	es	s	ns	es	s	ns			
0.0	1	1	1	1	1	0.99	1	1	1
0.1	1	1	1	1	0.96	0.96	1	0.92	0.93
0.2	1	1	0.99	1	0.88	0.88	1	0.77	0.76
0.3	1	0.99	0.97	1	0.74	0.75	1	0.51	0.45
0.4	1	0.90	0.91	1	0.59	0.54	0.80	0.24	0.18
0.5	1	0.79	0.83	0.8	0.36	0.28	0.40	0.08	0.04
0.6	1	0.62	0.64	0.4	0.17	0.13	0.20	0.02	0.0
0.7	1	0.36	0.34	0.4	0.05	0.05	0.0	0.0	0.0
0.8	0.6	0.10	0.07	0.4	0.01	0.01	0.0	0.0	0.0
0.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Avg.RDR	13.3	7	5.7	20	8.67	8	26.7	16.33	18.33

Table 5.14 – *The different recall values at different threshold value, th, and the average recall decrease rate (RDR) for exactly similar, es, similar, s, and nearly similar, ns, using SVS, LSA and VSM methods on the written part of the corpus annotated with the multi-category type annotation.*

90%. Even though the threshold at 0.4 seems to be good for the written part of our corpus, it may not be the case for other modalities present. To be sure of not missing out on the actual alignments with other modalities we should choose a smaller threshold. Selecting some smaller threshold will increase the probability of extracting most of the candidate alignment.

In table 5.13, we see that SVSM performs better than LSA and VSM in terms of the average recall decrease rate. This shows that SVSM is more stable in terms of the selection of threshold. This average recall decrease rate is also performs better with the corpus annotated with multi-category as shown in table 5.14. This property of decrease in recall is very important as we would like to use the threshold based on this experiment on other modalities of the corpus. With SVSM, we will be able to use it to select candidate alignments from other modality without the fear of loosing many actual alignments with an assumption that the actual alignments are similarly distributed in other modalities. In terms of the candidate alignments for the recall range 90%, SVSM extracts lesser candidate pairs compared to LSA. SVSM extracts about 8,546 pairs at 91% whereas LSA extracts 11,851 pairs at 96%. Even though VSM has a slightly lesser number of pairs in the range above 90% recall, we do not consider it because it is very discriminative which in turn may potentially leave out many actual alignments as it has a very low RDR.

5.6 Hybrid Method to Align Oral and Multimodal Corpus

In the previous section we showed that SVSM is able to select candidate alignments automatically. This automatic selection of candidate alignments will help the manual annotation process by reducing the annotation time and effort. We assume that the behaviour of SVSM along with other methods is similar on different modalities. Using this rep-

Hybrid method

resentation of text we are able to use hybrid method for the pair-wise alignment of a corpus. This hybrid method follows the two phase manual annotation process of the written text presented in the Section 5.4. In the first phase, the candidate alignments are automatically selected using the SVSM representation and in the second phase the actual alignments are manually selected from these candidate alignments. As the alignment of text in the corpus is done in 3 steps in which the 1st step of aligning the written text was done manually, and among the remaining two steps the 2nd step of intra-modal alignment between transcript texts and the 3rd step of inter-modal alignment between the written and the transcript texts will be aligned using this hybrid method.

5.6.1 Hybrid Alignment of Oral Corpus

First phase of Hybrid method

There are 403 transcript text segments which will produce a total of 81,003 text pairs for comparing. Function words are removed from these texts and the rest of the words were stemmed as done with the written part of the corpus. Using SVSM, the number of aligned pairs selected at each threshold is given in Table 5.15.

>Threshold	No. of Retrieved Pairs
0	77,823
0.1	65,789
0.2	49,164
0.3	32,664
0.4	20,592
0.5	12,560
0.6	6,587
0.7	2,349
0.8	410
0.9	31

Table 5.15 – *The number of retrieved pairs that were selected that could be used as candidate alignment pairs at different thresholds.*

As mentioned in the previous section, selection of threshold is not easy and to reduce the chances of missing out on the actual alignments we select the threshold of 0.3. This threshold was selected on the basis of the results obtained with the written text where at the threshold of 0.3 almost all the actual pairs were present in a significantly reduced set of candidate pairs as shown in 5.13. Using this threshold we selected 32,664 text pairs as our candidate alignments. Even though this number is still large, we were able to reduce the initial size by 48,339 text pairs which is slightly more than half of the initial text pairs.

Second phase of Hybrid method

The first phase of the alignment process is finished once we obtain our candidate alignment pairs. For the second phase, these candidate alignments are given to an annotator in seven groups for the manual selection of the actual alignments . Each group consists of text pairs with

the similarity value between a threshold range, i.e. the range from 1 to 0.3 threshold is divided in 7 parts as 1-0.9, 0.9-0.8 and so on. This division of text pairs was made to further help the annotator decide on similarity. The text pairs within a group will have relatively less variation in the nature of text pairs making the property of similarity and dissimilarity more vivid which in turn makes the decision making process quicker and easier.

Each of these groups are then given to the annotator, who is a linguist, and annotates each of them one group at a time. The candidate alignments present in the higher similarity range are few in number but they have a higher chance of being the actual alignments. These higher ranged groups concentrate the most possible similar pairs together making them visible to the annotator which prevents actual aligned pairs being unnoticed. The grouping also has a psychological factor in giving the annotator a break and a feeling of accomplishment which is important because the annotation process is mentally stressing as information keeps piling up as the texts are read which hinders the decision making process as information gets overlapped.

The annotator selects the actual alignment based on our alignment criteria of similarity with categorical annotations. The categories are *exactly similar*, *similar*, and *nearly similar* which indicates the fulfilment of the similarity criteria. The annotator was given 32,664 candidate alignments in the transcripts from which the actual alignments were selected. These selections were verified by two annotators who were involved in annotating the written part. After the selection process a total of 169 aligned text pairs were the actual alignments with 57 as exactly similar, 81 as similar, and 31 as nearly similar pairs. These are summarized in the table 5.16. The recall values of these different types of actual alignments with respect to the threshold is given in Table 5.17. This will give the distribution of the similar pairs along the threshold to elaborate the choice of the threshold and the difference of it with the written part.

Alignment phase	Initial pairs	Candidate pairs	Actual pairs	Exactly Similar pairs	Similar pairs	Nearly Similar pairs
1^{st}	81,003	32,664	-	-	-	-
2^{nd}	32,664	-	169	57	81	31

Table 5.16 – *Summary of the alignment process of the transcript text.*

Table 5.17 shows that the distribution of each category throughout the threshold range and there are some considerable amount of actual alignments below the 0.6 threshold. For the written part of our corpus, 0.4 threshold could be considered a good threshold which is able to separate candidate alignment pairs from the rest. This is slightly different with the transcript texts. It seems for the transcripts, the threshold at 0.5 would be appropriate. This could be due to the difference of the nature of the text. But because SVSM has a small average RDR value, this small difference do not produce a huge difference in the actual aligned pairs in the extracted

Thresholds	Types of Pairs			
	Exact Similar	Similar	Nearly Similar	All Similar
0.3	1.0	1.0	1.0	1.0
0.4	0.96	1.0	1.0	0.99
0.5	0.87	0.96	1.0	0.94
0.6	0.79	0.70	0.81	0.75
0.7	0.61	0.53	0.58	0.56
0.8	0.49	0.24	0.03	0.29
0.9	0.28	0.01	0.00	0.10

Table 5.17 – *The different recall values at different threshold of all the similar pair types using the SVSM method.*

candidate pairs.

5.6.2 Hybrid Alignment of Multimodal Corpus

The third and final step in the pair-wise annotation of the multimodal corpus is the alignment between the modalities, i.e. between 291 written text units and 403 transcribed text units. The same two phased hybrid method used for the transcribed text is used to select the actual alignments from the total of 1,17,273 total alignment pairs. The SVSM method at a threshold of 0.3 extracted 49,367 candidate alignments. Similar to the previous experiments, we removed function words and stemmed the content words while representing the text in SVSM. Table 5.18 shows the number of retrieved pairs as candidate alignments at different threshold levels.

>Threshold	No. of Retrieved Pairs
0	1,14,311
0.1	97,861
0.2	74,641
0.3	49,367
0.4	31,504
0.5	18,396
0.6	7,778
0.7	2,022
0.8	255
0.9	1

Table 5.18 – *The number of retrieved pairs that were selected for the transcript and written text alignment that could be used as candidate alignment pairs at different thresholds.*

From the extracted candidate alignments, an annotator manually extracted actual alignments which were verified by the same two annotators who verified the oral part. From this process a total of 125 actual alignments were extracted among which there are 34 exactly similar pairs, 68 similar pairs, and 23 nearly similar pairs. This alignment process is summarized in table 5.19.

The distribution of these similar pairs with respect to the recall can be seen in table 5.20. From the table we can see that a good threshold would

Alignment phase	Initial pairs	Candidate pairs	Actual pairs	Exactly Similar pairs	Similar pairs	Nearly similar pairs
1^{st}	1,17,273	49,367	-	-	-	-
2^{nd}	49,367	-	125	34	68	23

Table 5.19 – *Summary of the alignment process of the transcript and written texts.*

be 0.4. This is the same threshold of the written part and is very close to the oral part which was 0.5. The threshold 0.4 would be a good estimate for our multimodal corpora but across modalities, the recall values are different showing the difference in the properties and an indication of the sensitivity of choosing thresholds.

	Types of Pairs			
Thresholds	Exact Similar	Similar	Nearly Similar	All Similar
0.3	1.0	1.0	1.0	1.0
0.4	0.96	1.0	1.0	0.99
0.5	0.94	0.91	1.0	0.80
0.6	0.85	0.76	1.0	0.58
0.7	0.71	0.51	0.74	0.34
0.8	0.35	0.19	0.35	0.13
0.9	0.00	0.00	0.00	0.00

Table 5.20 – *The different recall values for the inter-modality alignment at different thresholds of all the similar pair types using the SVSM method.*

Conclusion

Building a new method that is able to automatically align similar texts require an annotated corpus for the test of the method. In this chapter we presented how a monolingual multimodal comparable corpus was created with alignments on the basis of similarity between short texts within this corpus. The corpus contained text from two modalities, i.e. written text and transcribed text. These texts related to the topic of the death of Diana and were collected and created from sources such as the Linguistic Data Consortium, ABC and CNN news cooperation.

To annotate any corpus, the text segment to annotate and the annotation or alignment criteria must be defined. In our corpus, we use the natural physical partitioning of the text, such as paragraph or turn markings, to get text segments. There are automatic methods that could do this but most of them have not been developed to extract short texts. These short texts are then aligned on the basis of the similarity criteria that states that if one of the main information is common between two text segments then they are similar. There are problems with this definition like finding the main information and the commonality between them. Despite this, the definition gives the freedom to the annotator to make a natural choice for deciding on similarity. This type of definition makes a binary decision on similarity. This is a bit rigid which leads to missing similar pairs. So, to make things flexible we proposed a modified criteria

which is implemented using categories which means we not only say that the pairs are similar but we also give the degree in which they are similar. We used exactly similar, similar, and nearly similar categories. These categories gives the annotator some guideline to follow and the rigidity of the binary decision is no longer present. This will ensure the different types of similar pairs to be present.

Having the short texts and the similarity criteria, we align the text pairs. The alignment was performed using a two phased manual method and the hybrid method where one of the phase in the manual method is automatized. The written text was first manually aligned using the two phased method. In the first phase the candidate alignment were selected. This selection took less analysis of the text as not much analysis is required. This phase produces a smaller set of text pairs that have the possibility of being actually similar. On this pool of text pairs we then manually select the actual alignment by analysing each pair.

We align the different modalities separately to divide the problem which saves time and human effort. We first align the written text using a two phase method. The first phase extracts the candidate alignments which are the alignments that include all the actual alignments and in the second phase the actual alignments were extracted from them manually. While extracting the actual alignments using the binary decision, we were able to save 75 hours of human effort compared to aligning the written part directly. The second phase was done again using the multi-category decision.

We further reduce the overall time and effort of the annotation by automatising the first phase of the manual process. We experimented with Vector Space Model (VSM) and Latent Semantic Analysis (LSA) to find the candidate alignments. LSA had a lower average RDR value than VSM but LSA still had a large number of candidate pairs in the range of 90% of recall. Due to these short comings we proposed a new similarity measure called Short Text Vector Space Model (SVSM). This takes into account the information of the importance of each term within the short texts, their co-occurrence with other terms, and the distribution of each term within the text to determine the similarity between text pairs. SVSM produced an average RDR of 2.87 which is much less than LSA and VSM. SVSM also has fewer candidate alignments in the range of 90% recall compared to LSA. This makes it a good choice for automatizing the first phase. Using this hybrid method we align the other modalities of the corpus. The threshold good for all the modalities seemed to be 0.4 but we selected 0.3 as a threshold to be sure to extract all the candidate alignments.

In the next chapter we experiment with different text representation methods and similarity methods to find a method that is capable of automatically extracting the similar pairs.

Multimodal Automatic Alignment

6

Contents

The task of alignment is to link texts based on some criteria. Alignment between texts can be done in a pair-wise or a group-wise manner. In the pair-wise alignment, text pairs that are similar are linked while in the group-wise alignment, all the texts that are similar to each other are linked together in a group. In this chapter, we present automatic methods for the pair-wise and group-wise alignment and evaluate their performance. The evaluation of these methods are done on the Multimodal Monolingual Comparable corpus which is the gold corpus built in chapter 5.

The automatic methods for pair-wise alignment are built using different combinations of text representations, weights, and similarity measures. Section 3.1 explains the way in which these three components combine to create an alignment method. The automatic alignment of texts is different from the automatic extraction of candidate alignment which was used for the hybrid alignment method presented in section 5.5. From the candidate alignments, the actual alignments are extracted manually to build the gold corpus whereas, the automatic alignment methods directly extract the actual alignments from the initial pool of texts. The gold corpus has been annotated with similar pairs, i.e. actual alignments, and these alignments are further pairs with further annotations on the category of similarity, i.e., exactly similar, similar, nearly similar, as mentioned

in section 5.4.2.

The differences between the automatic alignment and the extraction of candidate alignments are in terms of the *number of extracted alignments*, the *recall* and the *recall decrease rate* (RDR). For the candidate alignments, we require the recall to be very high within the extracted alignments without much care of the precision except that the amount of extracted alignments should be as small as possible. The most important point for the candidate alignment method is that the method should have a low recall decrease rate. This decrease rate insures that the loss of actual alignments is minimum. For the automatic alignment methods which directly extract the actual alignments, we would like a method that gives a high recall with as high precision as possible or in other words, we try to find the method which has a high f-score indicating a good balance between recall and precision.

The group-wise alignment also follows this idea of having a high recall and precision. Hence, the group-wise alignment methods will also be evaluated in terms of f-score but a modified version called maximum average f-score (MAF) to accommodate the different groups.

6.1 Pair-wise Alignment

6.1.1 Short texts

Pair-wise alignment between short texts is the process of making a link between two short texts based on their similarity. In section 5.5.2, we compared Vector Space Model (VSM), Latent Semantic Analysis (LSA) and Short text Vector Space Model (SVSM) to find the best method for the extraction of candidate alignments, i.e. pairs of text that have a possibility of being similar. In those experiments we found that SVSM performs better in finding candidate alignments from the initial pool of text pairs. Once the candidate alignments were extracted, we manually selected the actual aligned text pairs, which are similar text pairs. This manually selected pairs are the gold standard. Our goal is to find these actual alignments automatically from the initial pairs of short text. We use the same three text representation methods along with two other text representation methods namely Principle Component Analysis (PCA) and Independent Component Analysis (ICA) for the automatic alignment.

As automatic alignment method consists of a text representation method, similarity measure and weights, we evaluate 225 alignment methods from the combinations of 5 text representation methods, 3 similarity metric and 15 weights. Four of the text representation methods are Vector Space Model (VSM), Latent Semantic Analysis (LSA), Principle Component Analysis (PCA) and Independent Component Analysis (ICA) which were presented in chapter 3 and the fifth one is Short text Vector Space Model (SVSM) presented in section 5.5.2. The similarity metric are Cosine, Euclidean (more specifically Chord) and Jaccard similarity metric. Whereas the weights are the combination of local and global weights

presented in table 5.9 and table 5.10. These components are listed in table 6.1.

Text representation methods	VSM	SVSM	LSA	PCA	ICA
Similarity metrics	Cosine	Euclidean	Jaccard		
Weights	tf	tf*idf	tf*gwidf	tf*ent	tf*norm
	bintf	bintf*idf	bintf*gwidf	bintf*ent	bintf*norm
	logtf	logtf*idf	logtf*gwidf	logtf*ent	logtf*norm

Table 6.1 – *The different components that combine together to create automatic alignment methods.*

Experiments

The multimodal monolingual comparable corpus created in chapter 5 consists of written and transcribed texts and is used to evaluate the automatic methods. These texts have been segmented manually to create short texts. There are 240 written and 403 transcribed short text segments which gives a total of 28,680 and 81,003 text pairs respectively. The multi-modal part will combine these two modalities to have 643 text segments and will have a total of 96,720 (240 x 403) text pairs. Table 6.2 gives the number of actual alignments and the number of different types of similar pairs that are present in them. In each of the short texts, the function words were removed while the remaining terms were stemmed using the Snowball[1].

Modality	Segments	Initial pairs	Actual pairs	Exactly Similar pairs	Similar pairs	Nearly Similar pairs
Written	240	28,680	418	5	142	271
Transcript	403	81,003	169	57	81	31
Multimodal	643	96,720	125	34	68	23

Table 6.2 – *Summary of the multi-modal corpus showing the number of segments in the each modality and the total number of alignment pairs including their different categories.*

Like the manual annotation process in Section 5.4, the experiments for the automatic alignment is carried out one by one for each modality. We first extract the actual alignments from the written text, the transcribed text of the corpus separately, and then from between these two modalities. The extraction is carried out by selecting all the pairs with the similarity value greater than a threshold value as the actual alignment. The performance of all the methods is evaluated on each modality of the gold corpus using the f-score value. The f-score is measured at each threshold from 0 to 1 in an interval of 0.1. Out of the 225 alignment methods, Table 6.3 shows the highest f-score value reached by the text representation methods and their corresponding configuration, i.e. the weight, and similarity measure, to extract all the actual alignments, including exactly similar, similar and nearly similar pairs for each modality in the corpus.

[1]http://snowball.tartarus.org/

Written Text					
Representation	VSM	SVSM	LSA	PCA	ICA
Similarity	Cosine	Euclidean	Euclidean	Cosine	Euclidean
Weights	bintf	bintf*norm	logtf*idf	logtf*norm	tf*idf
Recall	0.56	0.44	0.40	0.3	0.38
Precision	0.32	0.40	0.40	0.44	0.5
F-score	0.41	0.42	0.40	0.35	**0.43**
Transcript Text					
Representation	VSM	SVSM	LSA	PCA	ICA
Similarity Measure	Jaccard	Cosine	Jaccard	Euclidean	Cosine
Weights	tf*idf	logtf*norm	tf*idf	tf*norm	tf*idf
Recall	0.25	0.37	0.22	0.21	0.31
Precision	0.55	0.32	0.63	0.25	0.38
F-score	0.34	**0.35**	0.32	0.23	0.34
Written-Transcript Text					
Representation	VSM	SVSM	LSA	PCA	ICA
Similarity Measure	Cosine	Euclidean	Cosine	Cosine	Cosine
Weights	logtf*idf	tf*norm	tf*norm	tf*norm	tf*idf
Recall	0.48	0.33	0.28	0.44	0.46
Precision	0.27	0.42	0.35	0.11	0.29
F-score	0.35	**0.37**	0.31	0.17	0.35

Table 6.3 – *The highest f-score value reached by the text representation methods on the actual pairs, which includes exactly similar, similar and nearly similar text pairs, along with the weight and similarity measure. The recall and precision values are also given for the best methods.*

In table 6.3, we can see that the best methods consisting each of the text representation methods along the same modality have similar maximum f-score value, except for PCA, but with some variations in the recall and precision value. Even though these maximum f-score values are similar across the same modality, the weights and similarity values used to achieve them are not. There is no one method that performs best across the modalities. These observations give hints on the importance of the combination of the text representation methods, similarity metric and the weights because with the proper combination, it is possible to achieve almost the same maximum f-score value with different combinations. Even though the maximum f-score can be reached using various combination of the aligning method components, this maximum value has a ceiling which is low indicating the difficulty of the problem.

As seen in the previous chapter, similarity is a complex idea and to understand this idea, we categorized similarity into three types, i.e. exactly similar, similar, and nearly similar. The different categories of similar pair have their own properties. We further evaluate our automatic methods on these different categories of similarity categories to understand more about the abilities and limitations of these methods. Table 6.4 gives the best methods with the highest f-score value for each similarity metric for Exactly Similar pairs. Similarly, table 6.5 and table 6.6 are related to Similar and Nearly Similar pairs respectively.

Written Text					
Representation	VSM	SVSM	LSA	PCA	ICA
Similarity Measure	Jaccard	Jaccard	Jaccard	Cosine	Jaccard
Weights	tf*gwidf	bintf*norm	logtf	tf*norm	tf
Recall	0.17	0.17	0.33	0.17	0.17
Precision	1.00	0.50	0.33	0.5	1.00
F-score	**0.29**	0.25	0.33	0.25	**0.29**
Transcript Text					
Representation	VSM	SVSM	LSA	PCA	ICA
Similarity Measure	Euclidean	Cosine	Jaccard	Euclidean	Cosine
Weights	bintf	tf*idf	tf*idf	tf*norm	bintf
Recall	0.51	0.53	0.53	0.33	0.51
Precision	0.62	0.65	0.51	0.43	0.62
F-score	0.56	**0.58**	0.52	0.38	0.56
Written-Transcript Text					
Representation	VSM	SVSM	LSA	PCA	ICA
Similarity Measure	Jaccard	Jaccard	Jaccard	Jaccard	Cosine
Weights	tf*idf	tf*norm	tf*norm	tf*norm	tf*gwidf
Recall	0.18	0.24	0.29	0.18	0.21
Precision	0.38	0.36	0.29	0.10	0.27
F-score	0.24	**0.29**	**0.29**	0.13	0.23

Table 6.4 – *The highest f-score value reached by the text representation methods on* ***Exactly Similar pairs*** *along with the weight and similarity measure. The recall and precision values are also given for the best methods.*

Written Text					
Representation	VSM	SVSM	LSA	PCA	ICA
Similarity Measure	Cosine	Euclidean	Cosine	Cosine	Euclidean
Weights	bintf	logtf*norm	tf*idf	logtf	bintf*gwidf
Recall	0.44	0.40	0.31	0.21	0.40
Precision	0.32	0.33	0.32	0.2	0.36
F-score	0.37	0.36	0.31	0.20	**0.38**
Transcript Text					
Representation	VSM	SVSM	LSA	PCA	ICA
Similarity Measure	Jaccard	Jaccard	Jaccard	Euclidean	Jaccard
Weights	tf*idf	tf*idf	tf*norm	tf*norm	tf*idf
Recall	0.46	0.26	0.17	0.10	0.42
Precision	0.09	0.14	0.11	0.06	0.09
F-score	0.14	**0.18**	0.14	0.07	0.15
Written-Transcript Text					
Representation	VSM	SVSM	LSA	PCA	ICA
Similarity Measure	Cosine	Euclidean	Jaccard	Cosine	Cosine
Weights	bintf*gwidf	bintf*norm	tf*norm	tf*norm	logtf
Recall	0.34	0.21	0.22	0.10	0.34
Precision	0.13	0.25	0.15	0.10	0.14
F-score	0.19	**0.22**	0.18	0.10	0.20

Table 6.5 – *The highest f-score value reached by the text representation methods on* ***Similar pairs*** *along with the weights and similarity measure. The recall and precision values are also given for the best methods.*

Even by breaking down the problem of similarity in categories, there is still uncertainty on the best method for each similarity category across the modalities. The complex nature of similar text even persists in specific categories of similarity. Despite this, there are text representation meth-

Written Text					
Representation	VSM	SVSM	LSA	PCA	ICA
Similarity Measure	Jaccard	Jaccard	Euclidean	Euclidean	Euclidean
Weights	tf	tf*idf	logtf*idf	logtf*gwidf	logtf*idf
Recall	0.35	0.38	0.44	0.11	0.39
Precision	0.27	0.27	0.2	0.01	0.27
F-score	0.31	**0.32**	0.27	0.02	**0.32**
Transcript Text					
Representation	VSM	SVSM	LSA	PCA	ICA
Similarity Measure	Euclidean	Jaccard	Cosine	Euclidean	Euclidean
Weights	tf*idf	bintf	logtf*norm	logtf*norm	tf*idf
Recall	0.48	0.52	0.26	0.65	0.58
Precision	0.03	0.04	0.03	0.01	0.03
F-score	0.05	**0.07**	0.05	0.02	0.05
Written-Transcript Text					
Representation	VSM	SVSM	LSA	PCA	ICA
Similarity Measure	Euclidean	Jaccard	Jaccard	Jaccard	Cosine
Weights	tf*gwidf	tf*norm	tf*norm	tf*norm	logtf
Recall	0.39	0.22	0.09	0.10	0.43
Precision	0.15	0.23	0.15	0.03	0.14
F-score	0.21	**0.22**	0.11	0.05	0.21

Table 6.6 – *The highest f-score value reached by the text representation methods on **Nearly Similar pairs**, along with the weight and similarity measure. The recall and precision values are also given for the best methods.*

ods that in majority perform better on certain texts. For instance, SVSM performs best while aligning transcribed text and multimodal text across the category of similarity and also in the written modality for the nearly similar pairs where as, ICA performs the best for the written modality across the categories of similarity. Similarly to the text representation, there is no one combination of similarity metric and weights that perform well across the modalities or categories of similarity. On the other hand, there are instances where two different weights may give the same result for a text representation method.

As explained in section 5.3 the problematic part of pair-wise short text alignment is that, even within short text, such as sentences, there may exist more than one topic being discussed and the alignment with another short text may depend on only one topic. This partial similarity does make the task in hand difficult. In section 6.2 we discuss the group-wise alignment of short texts which intends to align texts by grouping texts that are similar on the basis of the sub-topic it contains and evaluate different clustering algorithms for the purpose of alignment.

We have experimented with different automatic aligning methods on aligning short texts. To compare the effectiveness of these methods in another field of NLP which is not as scarce as short text alignment, we evaluate these methods to a well known problem of paraphrase alignment. This experiment is discussed in the next section.

6.1.2 Paraphrase Alignment

Paraphrase alignment is another type of pair-wise alignment where two sentences that express the same meaning with different words are linked. As paraphrase alignment is a subset of our problem, it would be a good comparison between our automatic alignment methods and the existing state of the art methods used in paraphrase alignment. We used the Microsoft Research Paraphrase Corpus(MSRPC) (Dolan et al. 2004) to compare different methods on the task of paraphrase detection. MSRPC consists of 5,801 pairs of sentences collected from a range of online newswire over a period of 18 months for experiments. This dataset is divided into 4,076 training pairs and 1725 test pairs. The training pairs consist of 3,900 paraphrases and the test pairs consist of 1,147 paraphrases. The remaining sentence pairs in the corpora are not paraphrases. We test the text representation methods of VSM, LSA, ICA, PCA and SVSM using the cosine similarity metric with weights of tf*idf on these test pairs and compare results with other methods which are tested on the same corpus. The evaluations of these methods are given in table 6.7.

Method	Threshold	Recall	Precision	F-measure
Proposed Method				
VSM	0.4	96.6	70.9	**81.8**
SVSM	0.6	95.9	70.4	81.3
LSA	0.6	97.7	69.8	81.5
PCA	0.4	97.7	69.8	81.5
ICA	0.4	98.8	69.2	81.4
Islam & Inkpen (2008) Corpus-based				
STS	0.6	89.1	74.7	81.3
Mihalcea et al. (2006) Corpus-based				
PMI-IR	0.5	95.2	70.2	81.0
LSA	0.5	95.2	69.7	80.5

Table 6.7 – *The recall and precision values of different methods based on the highest F-measure value on the MSRPC paraphrase corpus.*

In this table, the first section labelled Proposed Methods are the methods that we tend to evaluate. These methods do not use any external knowledge or data unlike the other two sections. The results from these two other sections are taken from Abdalgader and Skabar (2011) and are explained in 4.2.2. This table presents the best results according to the highest f-measure value achieved by increasing the threshold by 0.1. All our experiments were done with stems as terms and without stopwords as was done in the experiment of the short text alignments. The results for all the five text representation methods are comparable with the existing state of the art methods giving confidence on the performance of these automatic alignment methods.

6.2 Group-wise Alignments

In pair-wise alignment, the alignment links two text segments based on similarity. Here, the similarity depends on the common information that exist between the texts. As mentioned in section 5.3, the tricky part of pair-wise alignment is that, even within short texts, such as sentences, there may exist more than one information being discussed in it which tends to contain more than one sub-topic. The group-wise alignment try to capture the generality of common information that is used by the pair-wise alignment. In Section 4.2.4, the idea of a Group-wise alignment was presented. The basic idea of this alignment is to collect texts, that are related to the same sub-topic, in a group. This is done using several clustering algorithms presented in section 4.2.4. These algorithms are evaluated on the written part of the multimodal corpora.

Similar to the pair-wise alignment, the group-wise alignment needs to define the text units to be aligned and the alignment criteria. The alignment criteria exploits the similarity between sub-topics. As the grouping is done based on the similarity of topic and the possibility that a short text may be related to more than one sub-topic, we experiment with two different clustering methods. One is the hard or crisp clustering where there is no overlap between clusters or in other words a text can be related to only one sub-topic. The other clustering method is the soft or fuzzy clustering (Nock and Nielsen 2006) where the overlaps are possible which means a text can be related to more than one sub-topic.

6.2.1 Gold Standard

The gold standard for the group-wise alignment is created using the written part of the corpus which contains 240 short texts. Each of these texts was manually annotated with one of the 13 pre-defined categories, presented in table 6.8, depending on which category is closer to the text. The categories were manually created in order to generalize and capture all the different varieties of the texts by the analysis of the textual content. These categories are related to different information, for instance the chronological order, topics, events etc. and could be considered as sub-topics on the topic of death of Diana. This gold standard will be referred to as *written-Diana*.

For the hard clustering, each text unit is placed in at most one of the most likely categories. The annotations were done manually by two annotators and will act as the gold alignment against which the performance of the clustering methods will be tested. The reliability of agreement on the annotation of these categories according to kappa is 0.91, which is a good agreement. Some disagreement that arose was due to the fact that some text segments could be related to more than one category but had to be assigned to only one of them creating disagreements. The disagreements were resolved between the annotators by discussing the main idea of the texts and coming to an agreement. Table 6.8 gives the distribution of the paragraphs according to the categories and some other features of the

Categories	Paragraphs
Diana's life before accident	21
Driver's life before accident	5
Other's life before accident	9
Just before accident	18
Accident	10
Just after accident	22
Accident aftermath	8
Expression of grief	31
Funeral of Diana	46
Accusations	13
Cause of accident	17
Investigation	20
Media	20

Table 6.8 – *Distribution of short texts to the categories/sub-topics of the written-Diana corpus for* ***Hard/Crisp Clustering****.*

Feature	Value
Number of categories	13
Number of paragraphs	240
Total number of terms	5,526
Vocabulary size (terms)	1,761
Term average per text unit	23

Table 6.9 – *Features of written-Diana would like a corpus.*

written-Diana corpus is given in table 6.9.

For the soft clustering, the annotations made for the hard clustering were further enriched by assigning more categories to each text unit where appropriate. This additional annotation was done by one of the same annotators that provided the hard clustering annotations.

In total there were 30 text pairs that had an additional sub-topic annotation to them. Here below is one of the examples which had an addition of a sub-topic:

Though police again Monday declined to discuss their investigation into the fatal crash in a highway tunnel along the Right Bank of the Seine River, new details from witnesses emerged. Several described photographers swarming around the car just after the accident, taking pictures of the victims.

This text segment is annotated with the sub-topic *Just after accident* and *Investigation*. The part of the text segment that mentions "the police declined to discuss their investigation" is related to the *investigation* whereas the part which mentions the "swarming of photographers after the accident" is related to the topic *Just after accident*.

Categories	Paragraphs
Diana's life before accident	26
Driver's life before accident	6
Other's life before accident	9
Just before accident	18
Accident	12
Just after accident	22
Accident aftermath	15
Expression of grief	39
Funeral of Diana	47
Accusations	14
Cause of Accident	17
Investigation	25
Media	20

Table 6.10 – *The new text distribution after additional annotations for* ***Soft/Fuzzy Clustering****.*

6.2.2 Maximum Average F-Score Cluster Evaluation

Clusters can be evaluated using intrinsic or extrinsic information as explained in section 4.3.2. We will use the extrinsic information for the evaluation of clusters because we have a pre-defined gold standard as explained in section 6.2.1. Clustering F-score measure is one evaluation method that has been in frequent used but is not a reliable evaluation method (Amigó et al. 2009). There are several other extrinsic evaluation methods that give scores on the *quality* of the clusters and based on these scores we tend to decide on the appropriateness of the clustering method. Different evaluation methods have different properties (Amigó et al. 2009), so rather than selecting the best evaluation method we use a direct method which uses F-score values.

Clustering algorithms, that we will use, create a number of clusters equal to the number of categories/classes but they do not assign these clusters to their corresponds classes. We evaluate the quality of clusters by first mapping each cluster to a class and evaluate the quality of clusters with respect to the assigned class/category. This method assigns each cluster, generated by the clustering method, to a unique class/category in such a way that the average F-score (AF) for each pair of cluster and class is maximized called Maximum Average F-score (MAF) . F-score is defined with the value of $\beta = 1$ giving equal priority to the recall and precision. As we maximize the average F-score, the resulting pairs of cluster and class could be considered as the best practical solution. This is graphically illustrated in figure 6.1.

Maximum Average F-score **MAF**

A high value for MAF generally indicates a high level of agreement between the classes and the clusters. This optimal assignment is done automatically using the Hungarian Algorithm (Harold 1955). This algorithm takes in a F-score confusion matrix of n class against n clusters as

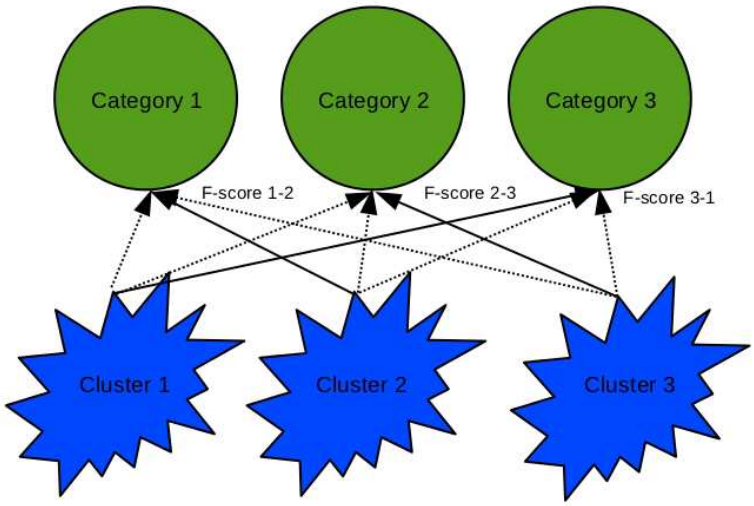


Figure 6.1 – *The clusters 1,2,3 are created by the clustering algorithm and categories 1,2,3 are the classes created manually. The lines, both solid and dotted, indicate the possible assignment of clusters to classes. The MAF evaluation assigns each cluster to a unique category such that the average F-score from the assignments is maximised. The assignment shown here, with a solid line, is possible if the F-score 1-2, F-score 2-3, and F-score 3-1 out of all possible assignments produce the maximum average F-score.*

shown in tables 6.12(a,b,c,d). The steps of the Hungarian Algorithm are summarized below :

1 Subtract the smallest entry in each row from all the entries of its row.

2 Subtract the smallest entry in each column from all the entries of its column.

3 Draw lines through appropriate rows and columns so that all the zero entries of the cost matrix are covered and the minimum number of such lines is used.

4 Test for Optimality: (i) If the minimum number of covering lines is n, an optimal assignment of zeros is possible and we are finished. (ii) If the minimum number of covering lines is less than n, an optimal assignment of zeros is not yet possible. In that case, proceed to 5.

5 Determine the smallest entry not covered by any line. Subtract this entry from each uncovered row, and then add it to each covered column. Return to 3.

6.2.3 Hard Clustering

Hard clustering is the clustering in which each text is assigned to only one class. This type of clustering is used in several applications, such as summarization, information retrieval and extraction. We use four types of clustering algorithm presented in section 4.2.4 which are, all three hierarchical clustering, i.e. Single Link, Complete Link, Average Link hierarchical clustering, and the spectral clustering methods. Each of these clustering methods require a dissimilarity or distance matrix created

using different similarity measures. These were generated by using two similarity measure, one the cosine similarity and the Kullback-Leibler distance metric. The cosine similarity measure is used with the VSM, LSA, and SVSM text representations whereas Kullback-Leibler distance is used only with VSM for the generation of the distance matrix. The weights used in the representation is the tf*idf weights for VSM and LSA, but for SVSM the weights were only the idf.

Unlike the situation for the pair-wise alignment, there are several freely available gold corpora for the group-wise alignment. To have a better understanding of the performances of different methods, we use four corpora to evaluate the clustering algorithms for hard clustering. These corpora are the written-Diana corpus which is the written part of the multimodal corpus, and three other corpora of different domain created to cluster short texts. These additional corpora are CICLing-2002 corpus, hep-ex corpus, and KnCr corpus of MEDLINE. A small descriptions of these corpora are presented below :

The CICLing-2002 corpus : This is a small corpus consisting of 48 abstracts in the domain of computational linguistics collected from the CICLing 2002 conference. This corpus has 4 classes of 48 abstracts and the abstracts are evenly distributed among the 4 classes which is as follows : {11, 15, 11, 11}.

The hep-ex corpus of CERN : This corpus contains 2,922 abstracts collected by the University of Jaén, Spain on the domain of Physics from the data server of the CERN. These abstracts are related to 9 categories. The distribution of the abstracts among the 9 classes is highly uneven and is as follows: {2623, 271, 18, 3, 1, 1, 1, 1, 1 }

The KnCr corpus of MEDLINE : This corpus contains abstracts from the cancer domain of the medical field and collected from the MEDLINE documents (Pinto and Rosso 2006). It contains 900 abstracts and they are related to 16 categories. The abstracts are distributed among the 16 classes as follows :{169, 160, 119, 99, 66, 64, 51, 31, 30, 29, 22, 20, 14, 12, 8, 6}

Table 6.11 shows the distributions of the texts from Cicling-2002 and hep-ex corpora among the clusters created by the four clustering algorithms. The similarity measure used in these clustering algorithms is the cosine similarity. This distributions of texts show that each clustering method has its own characteristics in terms of grouping texts which in turn define the type of clusters it creates.

From Table 6.11, we can see that SHC has the tendency of creating singleton clusters, the clsuters with only one element which indicates that it may not be a suitable choice. The characteristics of SPEC shows that it distributes the text evenly throughout the clusters. CHC and AHC have similar characteristics which lie between SHC and SPEC. These characteristics of the clustering method remain the same irrespective of

Cicling-2002 corpus

	Cluster Index			
Clustering	1	2	3	4
SHC	45	1	1	1
CHC	11	24	7	6
AHC	33	12	1	2
SPEC	13	4	9	22

hep-ex corpus

	Cluster Index								
Cluster	1	2	3	4	5	6	7	8	9
SHC	2912	1	1	1	1	1	1	1	1
CHC	2879	5	11	5	2	4	5	5	4
AHC	2879	13	11	1	5	3	5	2	1
SPEC	298	248	396	337	243	328	371	303	396

Table 6.11 – *Distribution of the text of text segments among the clusters created by SHC, CHC, AHC, and SPEC which uses cosine similarity*

(a) SHC

	Cluster			
Class	C1	C2	C3	C4
Cl1	**0.17**	0	0.36	0
Cl2	0	0	0.5	**0**
Cl3	0	0	**0.39**	0
Cl4	0	**0.17**	0.32	0.17

(b) CHC

	Cluster			
Class	C1	C2	C3	C4
Cl1	0.11	0.29	0.36	**0.12**
Cl2	0	**0.56**	0.15	0.19
Cl3	0	0.29	**0.45**	0.12
Cl4	**0.67**	0.17	0	0.24

(c) AHC

	Cluster			
Class	C1	C2	C3	C4
Cl1	0	0.17	0.36	**0.15**
Cl2	0	0.15	**0.54**	0
Cl3	**0**	0	0.5	0
Cl4	0.17	**0.70**	0.05	0.15

(d) AHC

	Cluster			
Class	C1	C2	C3	C4
Cl1	0	0.17	0.36	**0.15**
Cl2	0	0.15	**0.54**	0
Cl3	**0**	0	0.5	0
Cl4	0.17	**0.70**	0.05	0.15

(e) Cicling-2002

	F	ARI	V	MAF
SHC	0.40	0.01	0.11	0.21
CHC	0.52	0.10	0.21	0.45
AHC	0.53	0.17	0.29	0.35
SPEC	**0.61**	**0.25**	**0.34**	**0.60**

Table 6.12 – *In (a),(b),(c), and (d) the F-score confusion matrices for SHC, CHC, AHC, and SPEC applied on the CICLing-2002 corpus are shown and the elements which make the MAF are bold-faced. The classes and clusters are represented by the rows and columns respectively. In (e) the clusters generated by the clustering methods are evaluated using F, ARI, V, and MAF.*

the corpora. The best method for clustering cannot be decided based on the distribution of texts.

There are evaluation methods that give scores on the *quality* of the clusters and based on these scores we tend to decide on the appropriateness of the clustering method. The evaluation methods we use are F, ARI, and V which are explained in section 4.3.2, and Maximum Average F-score (MAF).

Tables 6.12(a,b,c,d) show the F-score confusion matrix of class against clusters generated by four clustering methods, using cosine similarity, on the Cicling-2002 corpus. The bold-faced values in each matrix makes the average F-score maximum. Table 6.12(e) shows the evaluation scores given to each clustering method by the 4 evaluation methods. We consider an evaluation method to be good if it resembles the MAF scores because a high value for MAF generally indicates a high level of agreement between the classes and the clusters.

Table 6.12 does not help us find the best evaluation method because no evaluation method represents the MAF value, but it certainly gives an insight on the performance of the clustering methods. All of the evaluation methods do point towards spectral clustering to be the best clustering method for our case. Table 6.13 gives the complete results of the experiments. It shows that for all the corpus, excluding hep-ex corpus, spectral clustering performs better than the rest. In the case of hep-ex, the short text are unevenly distributed among the clusters as presented in the description of the corpus. In contrast to the distribution in the hep-ex corpus, the characteristics of the spectral clustering tends to make evenly distributed clusters which explains its performance dealing with the hep-ex corpus.

For the hep-ex corpus, F evaluation method gives a good result for SHC even though the distribution of the short text in the clusters are clearly undesirable for other clusters as seen in Table 6.11. This is due to the drawback of F evaluation method. It is not may not take into account the membership of the clusters and may not evaluate the clusters. From this table we can also see that none of the evaluation measure resembles the MAF values. But if required, we would select V as the best out of the three evaluation methods. The reason behind this selection is that, V resembles the variation in the range of MAF more than the other evaluation measures. Among the 16 possible range of MAF, present in each box in Table 6.13, V resembles MAF 9 times where as ARI 7 times.

The MAF value is low for the written-Diana corpus. This value resembles the maximum f-score that was achieved in the pair-wise alignment of the written part of the corpus shown in table 6.3. This gives hints of the comparable difficulty between group-wise and the pair-wise alignment. But comparing the clustering algorithms, we consider spectral clustering to be a best clustering method among the four algorithms on the basis of the best evaluation scores achieved shown in Table 6.13. Spectral clustering achieves the best results with the LSA text representation method and the worst with KLD. This performance could be changed using different weights as was demonstrated during the pair-wise alignment in section 6.1.1. Further investigation into the effect of weights on spectral clustering should be carried out in the future.

6.2.4 Soft Clustering

In section 5.4, we mentioned that a short text may contain more than one information. We also pointed out that due to this, the alignment process tends to be difficult which in turn leads to the disagreements between annotators. While performing hard clustering, we annotated the text segments based on sub-topics and as there could be more than one information in one text, it is possible to have more than one sub-topic in each text segments. To handle such scenarios, where one text can be a member of more than one group, fuzzy or soft clustering is used. The fuzzy clustering is performed on the written-Diana corpus. The distribu-

Corpus		*KnCr*				*Cicling-2002*			
Cluster	Similarity	F	ARI	V	MAF	F	ARI	V	MAF
SHC	VSM	0.20	0.00	0.03	0.04	0.40	0.01	0.11	0.21
	KLD	0.20	0.00	0.03	0.04	0.40	0.01	0.11	0.21
	LSA	0.21	0.00	0.04	0.05	0.40	0.00	0.11	0.17
	SVSM	0.20	0.00	0.03	0.04	0.40	0.01	0.11	0.21
CHC	VSM	0.21	0.01	0.12	0.14	0.52	0.10	0.21	0.45
	KLD	0.20	-0.01	0.11	0.16	0.45	0.06	0.18	0.33
	LSA	0.21	0.03	0.12	0.09	0.52	0.11	0.23	0.52
	SVSM	0.22	0.01	0.09	0.10	0.46	0.07	0.19	0.40
AHC	VSM	0.25	0.04	0.12	0.13	0.53	0.17	0.29	0.35
	KLD	0.21	0.00	0.04	0.05	0.40	0.02	0.15	0.25
	LSA	0.21	0.00	0.06	0.07	0.40	0.00	0.10	0.21
	SVSM	0.20	0.00	0.04	0.04	0.40	0.02	0.15	0.25
SPEC	VSM	**0.30**	**0.09**	**0.19**	**0.19**	0.61	0.25	**0.34**	0.60
	KLD	0.23	0.04	0.11	0.14	0.51	0.15	0.26	0.51
	LSA	0.24	0.04	0.15	0.17	0.55	0.19	0.27	0.52
	SVSM	0.22	0.03	0.13	0.16	**0.64**	**0.26**	**0.34**	**0.64**
Corpus		*hep-ex*				*written-Diana*			
Cluster	Similarity	F	ARI	V	MAF	F	ARI	V	MAF
SHC	VSM	**0.86**	0.01	0.01	0.10	0.19	0.00	0.09	0.08
	KLD	**0.86**	-0.02	0.01	0.10	0.19	0.00	0.09	0.07
	LSA	**0.86**	0.01	0.01	0.11	0.19	0.00	0.09	0.07
	SVSM	**0.86**	0.01	0.01	0.11	0.19	0.00	0.09	0.08
CHC	VSM	**0.86**	-0.01	0.01	0.11	0.21	0.10	0.15	0.13
	KLD	0.81	-0.02	0.00	0.10	0.29	0.02	0.26	0.21
	LSA	0.41	0.01	0.02	0.08	0.29	0.08	0.28	0.25
	SVSM	0.56	0.03	0.07	0.11	0.41	0.24	0.42	0.24
AHC	VSM	**0.86**	0.03	0.01	0.11	0.38	0.18	0.38	0.21
	KLD	**0.86**	-0.01	0.00	0.10	0.35	0.14	0.36	0.18
	LSA	**0.86**	**0.10**	0.05	**0.13**	0.43	0.22	0.42	0.28
	SVSM	**0.86**	0.00	0.01	**0.13**	0.31	0.14	0.32	0.14
SPEC	VSM	0.28	0.01	**0.08**	0.09	0.50	**0.29**	**0.50**	0.41
	KLD	0.47	0.00	0.03	0.08	0.26	0.05	0.24	0.21
	LSA	0.28	0.01	**0.08**	0.09	**0.51**	0.27	0.49	**0.43**
	SVSM	0.29	0.01	**0.08**	0.09	0.45	0.23	0.45	0.36

Table 6.13 – *F,ARI, V, and MAF values for four clustering methods SHC, CHC, AHC and SPEC on four corpus KnCr, hep-ex,Cicling-2002, and the written part. The best score achieved by each evaluation method on every corpus are bold-faced. Here, cosine and KLD similarity measures are used on the VSM, LSA, and SVSM text representations. KLD uses VSM while cosine is used by VSM, LSA, and SVSM.*

tion of the texts across the categories are shown in table 6.10.

To perform fuzzy clustering, we use two algorithms, i.e. Fanny fuzzy clustering algorithm (Fanny) (Kaufman and Rousseeuw 1990) and Fuzzy c-means algorithm (FCM) (Bezdek et al. 1984). These clustering algorithms give a set of membership value to each short text, one for each category/class. The sum of the set of membership value is 1. Fanny aims to minimize the following objective function through an iterative process :

$$\sum_{v=1}^{k} \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} u_{iv}^{r} u_{jv}^{r} d(x_i, x_j)}{2 \sum_{j=1}^{n} u_{jv}^{r}} \tag{6.1}$$

where r is the membership exponent that controls the amount of fuzziness required and $d(x_i, x_j)$ is the dissimilarity between text x_i and x_j. Where as, u_{iv} is the unknown membership value of text i to cluster v and are subjected to the following constraints:

$$u_{iv} \geq 0 \text{ for } i = 1, .., n \text{ and } v = 1, ..., k \tag{6.2}$$

$$\sum_{v} u_{iv} = 1 \text{ for } i = 1, .., n \tag{6.3}$$

If $r \to 1$, clusters tend to have less overlap similar to the hard clustering where as if $r \to \infty$ the clusters have complete overlap making a very fuzzy clustering. The dissimilarity matrix from the written-Diana corpus is given by the Euclidean distance measure on the covariance matrix of the VSM text representation with weights as tf.

Similar to Fanny, FCM is based on minimizing the objective function by an iterative optimization process. For FCM, this function is as follows :

$$J_m = \sum_{i=1}^{n} \sum_{v=1}^{k} u_{iv}^{m} ||x_i - c_v||^2 \text{ , } 1 \leq m < \infty \tag{6.4}$$

where m is any real number greater than 1, u_{iv} is the degree of membership of x_i in the cluster v, x_i is the text i, c_v is the center of the cluster v, and ||*|| is any norm expressing the similarity between any text and the center. The dissimilarity matrix from the written-Diana corpus is given by the cosine similarity of the VSM text representation with weights as tf.

The iterative optimization of the objective function shown above, with the update of membership u_{iv} and the cluster centers c_v by:

$$u_{iv} = \frac{1}{\sum_{w=1}^{k} \left(\frac{||x_i - c_v||}{||x_i - c_w||} \right)^{\frac{2}{m-1}}} \text{ where, } c_v = \frac{\sum_{i=1}^{n} u_{iv}^{m} . x_i}{\sum_{i=1}^{n} u_{iv}^{m}} \tag{6.5}$$

This iteration will stop when $max_{iv}\{|u_{iv}^{k+1} - u_{iv}^{k}|\} < \epsilon$, where ϵ is a termination criterion between 0 and 1, whereas w are the iteration steps. This procedure converges to a local minimum or a saddle point of J_m.

As the soft clustering algorithm gives a set of membership values (MVs) to each text segment, each text is a member of all the possible classes but with different MVs. Because of this distribution of text across

all the classes, the evaluation of these clusters is not as straight forward as in hard clustering. A threshold has to be determined on these membership values in order to make each text a member of a smaller set of classes resembling the fuzzy annotation of the written-Diana corpus. For Fanny this threshold is $max(MV) - 0.005$ and for FCM it is $max(MV) - 0.0005$. These thresholds were determined empirically and are applied to the clusters created by the two clustering algorithms. The resulting new clusters were evaluated using the MAF evaluation method. Fanny produced an MAF value of **0.45** and FCM produced **0.38**. Fanny seems to be the better fuzzy clustering algorithm for our written-Diana corpus but there are many other factors that we have not investigated to confirm this result. For instance, the text representation methods, similarity measures, weights which corresponds to the input of the clustering algorithm whereas, there are elements within the clustering algorithm that may effect the performance of the algorithm such as the membership exponent value, number of iterations and so on.

The MAF values from the soft clustering and the hard clustering reach a similar value which tends to show that the quality of cluster is not much affected by the type of clusters created. This is an initial observation with the limited experiments that was carried out. More experiments with different combinations of components of the input similarity metric of the clustering algorithm and the parameters of the algorithm itself.

Conclusions

In this chapter, we experiment on different methods for the automatic alignments of short texts in a multimodal monolingual comparable corpus. The automatic alignment is different from the task of automatic extracting the candidate alignments presented in section 5.5 for the hybrid method. In the hybrid method, the candidate alignments were extracted automatically and the actual alignments were manually extracted from these candidate alignments. Whereas, in the automatic alignment, the actual alignments are directly extracted from the initial set of texts without any manual interference. The automatic alignment can be a pair-wise alignment or a group-wise alignment. In the pair-wise alignment each text pair are linked based on their information content whereas, group-wise alignment links all the texts that are similar to each other based on the sub-topics they contain.

In the pair-wise alignment, we use 225 methods which are the combinations of five text representation methods, three similarity method and 15 weights. These methods were applied to all the modalities that are present in the multimodal corpus. The experiments on the corpus show that the SVSM representation method performs the best for the transcribed text and the multimodal text whereas, for the written text ICA performs the best. Even though SVSM and ICA produce the best results for different modalities, the best values achieved by all the text representation method is close to one another, except for PCA. PCA does not seem to perform well in the context of alignment of short texts. There was no one method that was able to perform well across the modalities

but the performance of SVSM and ICA were consistent even in a more fine level where the different types of similarities were extracted separately. These experiments show that with the appropriate similarity measure and weights, the text representation methods perform similarly on our multimodal corpus. The highest f-score that any of the methods could reach was 0.43 on the actual alignments which is low indicating that the pair-wise alignment problem is difficult.

As these text representation methods have a low f-score value for the alignment task, we evaluated their performance on another similar NLP task called paraphrase detection. The performance of the representation methods in terms of the highest f-score value was comparable to the state of the art methods used in paraphrase detection. The best method was the VSM method but all the other text representations have the best f-score value close to it.

The group-wise alignment could be performed in two ways such that each text is assigned to only one group, called crisp clustering, or each text can be assigned more than one group, called fuzzy clustering. To evaluate the group-wise alignment we proposed a new way of evaluating clusters using an optimization algorithm and the F-score measure called the maximum average f-score (MAF). MAF assigns the cluster created by the clustering algorithm to a unique class such that the average F-score of each assignment is maximized. This optimized assignment is carried out using the Hungarian Algorithm.

Crisp clustering was done using four clustering algorithm among which Spectral clustering performed better compared to the other three hierarchical clusterings on four different corpora. Even though Spectral clustering performed the best, the maximum MAF value was 0.43 which was comparable to the pair-wise clustering indicating the same difficulty level. On the other hand, Fuzzy clustering was performed using Fanny fuzzy clustering and Fuzzy c-means clustering on the written part of multimodal corpus. Observing their MAF values, Fanny fuzzy clustering seems to perform better than Fuzzy c-means clustering with the MAF value 0.45 and 0.38 respectively. The MAF value again is similar to the pair-wise alignment and the crisp clustering showing that these three have similar difficulty levels. The crisp clustering and the fuzzy clustering both have not been thoroughly investigated compared to the pair-wise alignment. Further experiments have to carry out with different combinations of components and parameter settings for the group-wise alignment to understand their effects.

Conclusion and Future Work 7

The objective with which we started this thesis was to find an unsupervised automatic alignment method that would align short texts within texts generated from different modalities which are related to the same topic and are in one language. Short text alignments is a new subject with very few references. Alignments between short texts especially in the multimodal context is a hot topic in this golden age of information as it enables many applications to be built in the field of information retrieval/extraction and navigation. The challenge at the beginning was the alignment between modalities but in fact the difficult part was the alignment between short texts in general. Our aim has been to study the fundamental aspects of alignment and performances of different text representations on how well they extract the semantics of the texts for the purpose of alignment.

This thesis covers these aspects of the work on multimodal monolingual comparable corpus alignment. We focus on the adoption of unsupervised methods that do not require any training data and that use numeric approaches which do not require language specific linguistic resources. In line with these objectives, our main contribution includes the following:

1 Investigating the steps for alignment of short texts and developing a two phase manual method for alignment.

2 Presenting a new text representation method which will further reduce human effort and time for the creation of the gold corpus compared to the manual method.

3 Developing a multimodal monolingual comparable corpus.

4 Investigating the performance of different text representation methods on alignment and the effects of various weights and similarity measures.

5 Propose an external evaluation method to directly evaluate clusters using an optimization algorithm and the F-score measure.

The thesis is divided in two parts. The first part, which includes Chapter 2, 3 and 4, is dedicated to the scientific background and state-of-the-art whereas, the second part of the thesis includes Chapter 5 and 6 which is the heart of the thesis. In chapter 2 we first explain the concepts that deal with monolingual textual alignment and later present existing methods on text representation, segmentation, and alignment components. Section

2.1 explains the concept of a corpus including multimodal monolingual comparable corpus and the alignment process. We present two types of alignment, i.e. pair-wise and group-wise alignment, both of which are followed in our work. We view the alignment problem as an arrangement problem therefore, we consider linking two text pairs as well as grouping several text segments as a possibility of arranging texts. In Section 2.2, we present the overview of the short text units and alignment criteria which are two important concepts of alignment. We explain the three ways of generating or identifying short text boundaries using manual, naive and automatic methods. This section goes further to introduce the alignment criteria which is similarity. It explains the subjective nature of similarity which is the reason that makes it difficult to define and even with a definition, identifying the similar texts is difficult. These issues of text units and alignment criteria are the basis of alignment and their variability and problematic aspects are shown in this section to understand the problem in hand.

In Chapter 3, we explain how text in general can be represented in terms of vectors, terms and their weights such that the semantics of texts are mathematically expressed. We present the Vector Space Model (VSM) and three other text representation methods. These methods are Latent Semantic Analysis (LSA), Principle Component Analysis (PCA), and Independent Component Analysis (ICA). Each of these representations are mathematical transformations of VSM and tries to improve the representation of texts. The formal and mathematical foundations of these representations are presented in Section 3.1. In Section 3.2 we present several similarity measures which are used to assign a value of similarity between texts using the text representations. These two parts are essential to the automatic alignment process.

Besides text representations, there are two important parts in textual alignment which is the segmentation process and the alignment process. In Chapter 4, we present the state-of-the-art for these two processes. As presented in Chapter 2, segmentation can be done in three ways, manually segmenting the text for short texts, taking the physical partitioning of the text as the short text segmentation, and finally performing segmentation automatically. In Section 4.1 we focus on existing methods for automatic segmentation. These automatic processes rely on different features of the text. Among these methods, statistical methods do tend to perform better but a direct comparison of different methods only based on the published results is difficult as each application of segmentation is different. However, TextTiling and C99 are two methods for segmentation which are mostly taken as the baseline to compare new methods.

There are very few studies done in the field of short text alignment both in pair-wise and group-wise alignments. Among them, in section 4.2 we presented the work of Hatzivassiloglou et al. (2001) who works on alignment of paragraphs which is the study that is directly related to our aim. Their work focuses on different linguistic features which are co-occurrence, noun phrase matching, synonyms and overlap of common

verb classes and proper nouns. Their result show some improvement on the basic methods of tf*idf methods but with a low f-measure. This shows that the alignment problem is difficult and more deep studies have to be made to be able to use these linguistic features or some other features that can help alignment. We also present other related works which are sentence alignment proposed by (Barzilay and Elhadad 2003) and Nelken and Shieber (2006) and sentential paraphrase alignment by Mihalcea and Corley (2006), Barzilay and McKeown (2001), Islam and Inkpen (2008). Along with the pair-wise alignment, we also present some studies made on group-wise short text alignment by Makagonov et al. (2004), Pinto et al. (2007) and Pinto and Rosso (2006) and several state-of-the-art clustering algorithms. For the evaluation of these pair-wise and group-wise alignment methods we present in Section 4.3 the standard methods to evaluate results of both pair-wise and group-wise alignments.

Chapter 5 is the centrepiece of this thesis. The first three sections are focused on the basis of alignment but in contrast to Chapter 2, it gives specific descriptions which explains our work and context. In Section 5.1 we present the corpus and its structure that we collected to evaluate our work. Section 5.2 explains the difficulty in using automatic segmentation methods for our purpose and why we chose to select the physical structures of the texts as segmentation boundaries as well as the manual segmentation. The alignment criteria for our task is explained in Section 5.3. In our work we follow the intuitive similarity definition which states that, *Two text segments are similar if they contain at least one common main information*. This definition of similarity will always be behind our alignment process.

In addition to these fundamental parts of alignment, we explain how a gold corpus can be created in a way that human effort and time can be saved compared to traditional manual creation in Section 5.4. Due to the lack of a gold corpus to evaluate alignment methods, we started out on the creation of a gold corpus. We explain and present details that are usually either ignored or taken for granted. Issues such as the differences in the way short texts could be similar and properties of similarity which could be exploited to reduce the human time and effort for manual annotations. The possibility of several informations in a text which are distributed and interlinked throughout the text make the identification of similar text very tricky. To overcome this problem, we divided the manual alignment problem in two, creating a two phase alignment process. The first phase extracted the candidate alignments and the second phase was selecting the actual alignments from the candidate alignments based on the similarity criteria. The agreement between the annotators on selecting the actual alignments was 0.5 Kappa value indicating the difference on how different people perceive similarity. Even with this two phase method we still required a manual effort and to reduce the manual effort, in Section 5.5 we present a hybrid method which is still a two phase alignment process but the first phase of the alignment process is automated using a new text representation method called the Short text Vector Space Model (SVSM). This method drastically reduces the time of annotation from 91

hours to approximately 20 hours and also show that SVSM performs better than LSA and VSM in terms of the average recall decrease rate (RDR) and the extraction of the candidate pairs for the second phase. SVSM is also better compared to other types of text representations presented in Chapter 3 and can be seen in the Appendix A.1. With the two phase alignment process, different ways in which texts are similar were evident which enabled us to make sub-categories of the similarity definition in terms of exactly similar, similar and nearly similar text pairs. This break down of the similarity definition gave a clearer view on the degree of fulfilment of the similarity between texts but produced more similar text pairs compared to the previous definition. This increase is due to a better guideline of identifying similar pairs. The annotation on the actual alignments using these sub-categories produced more clarity in understanding similarity but also increased the disagreements between annotators with a lesser Kappa of 0.41. Using this SVSM method we align the rest of the multimodal corpus and show the results in Section 5.6.

Finally, after the creation of the gold corpus we present the automatic alignment between short texts in Chapter 6. We try to find the best text representation method for extracting the semantics of the text. We start by the automatic pair-wise alignment of short texts in using the different text representations in Section 6.1. The results of these alignments are given in two parts. The first part presents the automatic alignment of all the actual alignments including all three sub-categories of similar pairs. Whereas in the second part, the automatic alignment of each of the sub-category is performed separately. Comparing the results of these automatic alignments, we observe that there is no one text representation method that performs well across modalities but if a winner is to be chosen, then within modalities, in most of the cases, SVSM and ICA perform slightly better. Even though they are better, within each modality the maximum f-score value reached by all the text representations are close to each other and are attained with different weights and similarity measures. This shows the importance of the selection of an appropriate similarity measure and weight which has been pointed out by Nakov et al. (2001) and Dumais et al. (1998). Besides focusing on the text representation, the results presented in Chapter 6 also expresses the scale of difficulty in extracting the different sub-categories of similar pairs. Across the sub-categories and along each modality, nearly similar texts are more difficult to extract than the exactly similar pairs. With our limited study, we are not able to understand the nature of the results produced by each text representations as the link between semantics within short texts and the mathematical representation of the methods have not been completely understood. The maximum f-score value of 0.43 could only be reached while extracting all the actual alignments across the modalities. This value is low and so we try our text representation methods on paraphrase alignment which is a sub-task of short text alignment. The text representation methods performed slightly better than the state-of-the-art methods. This gives an idea of the difficulty present in the pair-wise alignment task.

Another aspect of short text alignment is the group-wise alignment

and is presented in Section 6.2. The group-wise alignment can be performed using a hard or soft clustering algorithm. The hard clustering can be assumed to be an easier problem compared to the pair-wise alignment because the agreement between the annotators while annotating the written part of the corpus was very good with a Kappa of 0.91 indicating there are no intuitive differences in the sub-categories unlike in the pair-wise alignment. There are several evaluation methods to evaluate different clustering algorithms but due to the different properties of these methods we proposed a new extrinsic evaluation method for the clustering problem called the Maximum Average F-score (MAF). This evaluation method uses the F-score measure and the Hungarian optimization algorithm (Harold 1955) for easy interpretation of the evaluation score. With the help of this evaluation measure we evaluated different hierarchical clustering and spectral clustering algorithms for hard clustering of the written part of the multimodal corpus along with other three gold corpora for clustering. In terms of MAF, spectral clustering performs better across all the corpora and for our corpus a value of 0.43 was achieved. In terms of soft clustering we used Fanny and Fuzzy c-means algorithm and they produced a MAF value of .45 and .38 indicating Fanny producing better homogeneous clusters than Fuzzy c-means algorithm. Even though the kappa of the group-wise alignment is high, the problem of alignment seems to be equally difficult as the pair-wise alignment.

We explored different methods to achieve automatic alignment and it is clear that this is not a solved problem. More research will be required, empirical and linguistic, to understand how the main information could be decrypted from short texts. This will lead to a better understanding of our results that we presented in Chapter 6. We could reach only so much f-score value with all the combination of different text representations, weights and similarity measures. Other than the ones we used, there are a vast spectrum of weights, text representations, similarity measures, clustering algorithms that have to be investigated in order to find a better mix of these elements. On the other hand, there should be more focus on the segmentation of texts itself. As we align texts in an information level, the need to design text segmentation methods at the information unit should also be a priority. For simplicity reason, we take the natural physical segmentation as boundaries for text units with some manual segmentation but these text units should be revised in order to achieve alignment in the information level.

In future work, beside continuing on empirical experiments, linguistic aspects of alignment has to be studied mostly on the level of information which is in line of the continuation of the work of Hatzivassiloglou et al. (2001). Our experiments give some hint that mathematical tools alone may not be enough for the alignment process and that eventually some NLP tools will be used for better extraction of the semantics of the texts. Some NLP topics dealing with the coherence of texts like Discourse entities (Recasens et al. 2013), Named Entity Recognition (M et al. 2013), and Anaphora resolution (Jagan et al. 2012) should be studied

for their use in the extraction of semantics of text in the alignment process.

Short text alignment process contains many intermediate steps that new as well as old hypothesis and assumptions have to be tested under a wide range of conditions. This thesis has touched and studied some of the important aspects of short text alignment but still there is a long way to go before alignment will be a solved problem.

List of Figures

List of Tables

Appendixs

A

A.1 Appendix A

This Appendix is the continuation of Chapter 5.

A.1.1 Hybrid Alignment

In section 5.6, we presented the hybrid method for the pair-wise alignment of the actual alignments. Even though the hybrid method does not extract the actual alignments automatically, it automatically extracts candidate alignments. These candidate alignments are text pairs that are possible to be actual alignments. This automatic step was performed using SVSM, LSA, and VSM. Among these methods SVSM was the method that was the most suitable for extracting the candidate alignments because it had the lowest average recall decrease rate (RDR) and the number of candidate alignments were low as well.

The new text representation methods, PCA and ICA has not been evaluated for this task. In table A.1, the combination of the text representation, similarity measure and weights are presented that produced the least average RDR for SVSM, PCA and ICA.

Best Method	SVSM		PCA		ICA	
Metric	Cosine		Cosine		Cosine	
Weight	tf		logtf * norm		tf * entropy	
Threshold	Retrieved	Recall	Retrieved	Recall	Retrieved	Recall
0.0	28680	1	5627	0.93	15989	1
0.1	25777	1	195	0.35	7370	0.96
0.2	18718	0.99	45	0.16	2373	0.81
0.3	13160	0.98	8	0.04	664	0.51
0.4	8546	0.91	2	0.01	174	0.24
0.5	4420	0.82	1	0.01	40	0.08
0.6	1779	0.64	1	0.01	6	0.03
0.7	498	0.35	0	0	0	0
0.8	66	0.09	0	0	0	0
0.9	0	0	0	0	0	0
Average recall decrease rate	2.87		29.63		16.20	

Table A.1 – *The SVSM representation, along with PCA and ICA, with the weight and similarity metric that produces the least average decrease rate in recall with three different similarity metric on the written part of the corpus annotated using the binary type annotation.*

From the table we see that SVSM still has the least average RDR value compared to PCA and ICA. PCA and ICA have a high average RDR value indicating they are both discriminative. Even though they both retrieved less candidate alignment with the recall greater than 90, they will not be good for extracting candidate alignments because with a small change in the threshold, the number of actual alignments are reduced drastically which indicates that SVSM is the best choice to extract the candidate alignments in the hybrid method of alignment.

Bibliography

Khaled Abdalgader and Andrew Skabar. Short-text similarity measurement using word sense disambiguation and synonym expansion. *AI 2010:Advances in Artificial Intelligence, Lecture Notes in Computer Science*, 6464:435–444, 2011. (Cited on page 107.)

Jens Allwood. Multimodal corpora. In *Corpus Linguistics. An International Handbook*, pages 207–225, 2008. (Cited on page 9.)

Enrique Amigó, Julio Gonzalo, and Javier Artiles. A comparison of extrinsic clustering evaluation metrics based on formal constraints technique. In *Information Retrieval*, page 261–286, 2009. (Cited on pages 69 et 110.)

Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*, 12:461–486, 2009. ISSN 1386-4564. (Cited on page 110.)

Artstein and Massimo Poesio. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34:555–596, 2008. (Cited on page 86.)

Ricardo Baeza-yates and Berthier Ribeiro-Neto. Modern information retrieval, 1999. (Cited on page 50.)

Daniel Bär, Torsten Zesch, and Iryna Gurevych. A reflective view on text similarity. In *Proceedings of Recent Advances in Natural Language Processing*, pages 515–520, 2011. (Cited on page 18.)

Alberto Barron-Cedeno, Andreas Eiselt, and Paolo Rosso. Monolingual text similarity measures: A comparison of models over wikipedia articles revisions. In *Proceedings of the 7th International Conference on Natural Language Processing*, pages 29–38, 2009. (Cited on page 18.)

Alberto Barrón-Cedeño, Martin Potthast, Paolo Rosso, Benno Stein, and Andreas Eiselt. Corpus and Evaluation Measures for Automatic Plagiarism Detection. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh conference on International Language Resources and Evaluation*, 2010. ISBN 2-9517408-6-7. (Cited on pages 10 et 54.)

Regina Barzilay and Noemie Elhadad. Sentence alignment for monolingual comparable corpora. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 25–32, 2003. (Cited on pages 1, 8, 20, 55, 57, 58 et 121.)

Regina Barzilay and Kathleen R. McKeown. Extracting paraphrases from a parallel corpus. *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 50–57, 2001. (Cited on pages 55, 58, 59 et 121.)

Regina Barzilay, Kathleen McKeown, and Michael Elhadad. Information fusion in the context of multi-document summarization. In *Association for Computational Linguistics*, 1999. (Cited on page 62.)

Jörg Becker and Dominik Kuropka. Topic-based vector space model. *Proceedings of the 6th International Conference on Business Information Systems*, pages 7–12, 2003. (Cited on page 31.)

Doug Beeferman, Adam L. Berger, and John D. Lafferty. Statistical models for text segmentation. *Machine Learning*, 34(1-3):177–210, 1999. (Cited on page 53.)

Silvia Bernardini. Monolingual comparable corpora and parallel corpora in the search for features of translated language. In *SYNAPS - A Journal of Professional Communication*, 2011. (Cited on page 8.)

Michael W. Berry, Susan, and T. Dumais. Using linear algebra for intelligent information retrieval. *SIAM Review*, 37:573–595, 1995. (Cited on page 32.)

J. Bezdek, R. Ehrlich, and W. Full. FCM: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, 10(2-3):191–203, 1984. (Cited on page 116.)

David M. Blei and Pedro J. Moreno. Topic segmentation with an aspect hidden markov model. pages 343–348. ACM Press, 2001. (Cited on page 54.)

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003. ISSN 1532-4435. (Cited on page 34.)

Eric Gaussier Bo Li. Improving corpus comparability for bilingual lexicon extraction from comparable corpora. *Proceedings of 23rd international conference on computational linguistics*, pages 644–652, 2010. (Cited on page 7.)

Gemma Boleda, Stefan Bott, Rodrigo Meza, Carlos Castillo, Toni Badia, and Vicente López. Cucweb: a catalan corpus built from the web. *Proceedings of the 2nd International Workshop on Web as Corpus*, pages 19–28, 2006. (Cited on page 6.)

John Broglio, James P. Callan, W. Bruce Croft, and Daniel W. Nachbar. Document retrieval and routing using the inquery system. In *Proceeding of Third Text Retrieval Conference (TREC-3*, pages 29–38, 1994. (Cited on page 28.)

Chris Buckley. The importance of proper weighting methods. *Workshop on Human Language Technology*, pages 349–352, 1993. (Cited on page 27.)

Marc Caillet, Jean-François Pessiot, Massih-Reza Amini, and Patrick Gallinari. Unsupervised learning with term clustering for thematic segmentation of texts. In *RIAO*, pages 648–657, 2004. (Cited on page 51.)

Maria Fernanda Caropreso, Maria Fernandacnandad, Stan Matwin, and Fabrizio Sebastiani. A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization. In *Text Databases and Document Management: Theory and Practice*, 2001. (Cited on pages 26 et 27.)

Yun-Chuang Chiao and Pierre Zweigenbaum. Looking for candidate translational equivalents in specialized, comparable corpora. In *International Conference on Computational Linguistics*, 2002. (Cited on page 6.)

Rudi L. Chilibrasi and Paul M. B. Vitanyi. The google similarity distance. In *IEEE Transactions on Knowledge and Data Engineering*, pages 370–383, 2007. (Cited on page 60.)

Freddy Y. Y. Choi. Advances in domain independent linear text segmentation. In *ANLP*, pages 26–33, 2000. (Cited on pages 48 et 77.)

Freddy Y. Y. Choi, Peter Wiemer-Hastings, and Johanna Moore. Latent semantic analysis for text segmentation. In *Proceedings of Empirical Methods in Natural Language Processing*, pages 109–117, 2001. (Cited on pages 32 et 49.)

Vincent Claveau. Vectorisation, okapi et calcul de similarité pour le tal : dpour oublier enfin le tf-idf. In *Proceedings of JEP-TALN-RECITAL*, pages 85–98, 2012. (Cited on page 28.)

William W. Cohen. Learning trees and rules with set-valued features. pages 709–716, 1996. (Cited on page 61.)

Michael Collins and Yoram Singer. Unsupervised models for named entity classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 100–110, 1999. (Cited on page 59.)

Robert Dale. Classical approaches to natural language processing. In Nitin Indurkhya and Fred J. Damerau, editors, *Handbook of Natural Language Processing, Second Edition*, Boca Raton, FL, 2010. CRC Press, Taylor and Francis Group. ISBN 978-1420085921. (Cited on page 13.)

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, (6):391–407, 1990. (Cited on pages 32, 33 et 93.)

Bill Dolan, Chris Quirk, and Chris Brockett. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. *20th International Conference on Computational Linguistics*, pages 350–356, 2004. (Cited on pages 10, 12, 60 et 107.)

David Dubin. The most influential paper gerard salton never wrote. *Library Trends*, 2004. (Cited on page 30.)

Susan Dumais, John Platt, Mehran Sahami, and David Heckerman. Inductive learning algorithms and representations for text categorization. In *Proceedings of CIKM-98, 7th ACM International Conference on Information and Knowledge Management*, pages 148–155. ACM Press, 1998. (Cited on pages 27 et 122.)

Susan T. Dumais. Improving the retrieval of information from external sources. In *Behavior Research Methods, Instruments, and Computers*, pages 229–236, 1991. (Cited on page 29.)

Olivier Ferret. Finding document topics for improving topic segmentation, 2007. (Cited on page 50.)

William B. Frakes and Ricardo A. Baeza-Yates, editors. *Information Retrieval: Data Structures & Algorithms*. Prentice-Hall, 1992. ISBN 0-13-463837-9. (Cited on page 50.)

Norbert Fuhr and Chris Buckley. A probabilistic learning approach for document indexing. *ACM Transactions on Information Systems*, 9:223–248, 1991. (Cited on page 26.)

Benjamin C.M. Fung, Ke Wang, and Martin Ester. Hierarchical document clustering using frequent itemsets. In *Proceedings of SIAM International Conference on Data Mining*, 2003. (Cited on page 66.)

Pascale Fung. A statistical view on bilingual lexicon extraction: From parallel corpora to non-parallel corpora. In *AMTA*, pages 1–17, 1998. (Cited on page 6.)

Pascale Fung and Kenneth Ward Church. K-vec: A new approach for aligning parallel texts. In *International Conference on Computational Linguistics*, pages 1096–1102, 1994. (Cited on page 6.)

Robert Gaizauskas, Jonathan Foster, Yorick Wilks, John Arundel, Paul Clough, and Scott Piao. The meter corpus: A corpus for analysing journalistic text reuse. pages 214–223, 2001. (Cited on pages 10 et 54.)

Luigi Galavotti. Experiments on the use of feature selection and negative evidence in automated text categorization. In *Proceedings of ECDL-00, 4th European Conference on Research and Advanced Technology for Digital Libraries*, pages 59–68. Springer Verlag, 2000. (Cited on page 27.)

William A. Gale and Kenneth W. Church. A program for aligning sentences in bilingual corpora. pages 1–8, 1991. (Cited on page 58.)

Michel Galley, Kathleen McKeown, Eric Fosler-Lussier, and Hongyan Jing. Discourse segmentation of multi-party conversation. In *Association for Computational Linguistics*, pages 562–569, 2003. (Cited on pages 47, 53 et 54.)

Peter Gärdenfors. Conceptual spaces : The geometry of though. MIT Press, 2000. (Cited on page 18.)

Mike Thelwall Georgios Paltoglou. A study of information retrieval weighting schemes for sentiment analysis. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 2010. (Cited on page 28.)

Nelson Goodman. *Seven strictures on similarity*. Bobbs-Merrill, 1991. (Cited on page 20.)

Sylviane Granger. Comparable and translation corpora in cross-linguistic research. design, analysis and applications. In *Journal of Shanghai Jiaotong University*, 2010. (Cited on page 7.)

B. J. Grosz and C. L. Sidner. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12:175–204, 1986. (Cited on page 18.)

Camille Guinaudeau, Guillaume Gravier, and Pascale Sébillot. Enhancing lexical cohesion measure with confidence measures, semantic relations and language model interpolation for multimedia spoken content topic segmentation. In *Journal of Computer Speech and Language*, pages 90–104, 2012. (Cited on pages 52, 54 et 77.)

Weiwei Guo and Mona Diab. Modeling sentences in the latent space. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 864–872, Jeju Island, Korea, 2012. Association for Computational Linguistics. (Cited on page 34.)

M. A. K. Halliday and Ruqaiya Hasan. *Cohesion in English*. Longman Group Limited, 1976a. ISBN 978-0-582-55041-4. (Cited on pages 14, 46, 47 et 50.)

Michael Alexander Kirkwood Halliday and Ruqaiya Hasan. *Cohesion in English*. Longman, London, 1976b. (Cited on pages 91 et 92.)

Khun W. Harold. The hungarian method for the assignment problem. In *Naval Research Logistics Quarterly*, volume 2, pages 83–97, 1955. (Cited on pages 110 et 123.)

Vasileios Hatzivassiloglou, Judith L. Klavans, and Eleazar Eskin. Detecting text similarity over short passages: Exploring linguistic feature combinations via machine learning, 1999. (Cited on pages 13, 14, 20 et 82.)

Vasileios Hatzivassiloglou, Judith L. Klavans, Melissa L. Holcombe, Regina Barzilay, Min yen Kan, and Kathleen R. McKeown. Simfinder: A flexible clustering tool for summarization. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Workshop on Automatic Summarization*, pages 41–49, 2001. (Cited on pages 1, 20, 55, 56, 58, 60, 120 et 123.)

Marti A. Hearst. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23:33–64, 1997. (Cited on pages 46, 77 et 92.)

Sanjika Hewavitharana and Stephan Vogel. Enhancing a statistical machine translation system by using automatically extracted parallel corpus from comparable sources. *Proceedings of the LREC 2008 Workshop on Comparable Corpora*, 2008. (Cited on page 7.)

Michael Hoey. *Patterns of Lexis in Text*. Oxford University Press, New York, 1991. (Cited on page 91.)

Thomas Hofmann. Probabilistic latent semantic analysis. In *UAI*, pages 289–296, 1999. (Cited on page 34.)

Timo Honkela, Aapo Hyvärinen, and Jaakko J. Väyrynen. Wordica - emergence of linguistic representations for words by independent component analysis. *Natural Language Engineering*, 16:277–308, 2010. (Cited on pages 26, 35 et 38.)

Anna Huang. Similarity measures for text document clustering. In Jay Holland, Amanda Nicholas, and Delio Brignoli, editors, *New Zealand Computer Science Research Student Conference*, pages 49–56, 2008. (Cited on pages 40 et 41.)

Lewis Hubert and Papa Arabie. Comparing partitions. In *Journal of Classification*, volume 2, pages 193–218, 1985. (Cited on page 67.)

W. John Hutchins. Machine translation: a concise history. 2007. (Cited on page 9.)

Aapo Hyvärinen. Survey on independent component analysis, 1999. (Cited on pages 36 et 38.)

Aminul Islam and Diana Inkpen. Semantic text similarity using corpus based word similarity and string similarity. *ACM Transactions on Knowledge Discovery from Data*, 2, 2008. (Cited on pages 55, 58, 60 et 121.)

Aminul Islam, Diana Inkpen, and Iluju Kiringa. Applications of corpus-based semantic similarity and word segmentation to database schema matching. *The VLDB Journal - The International Journal on Very Large Data Bases*, pages 1293–1320, 2008. (Cited on pages 1 et 60.)

Balaji Jagan, T V Geetha, and Ranjani Parthasarathi. Two-stage bootstrapping for anaphora resolution. In *Proceedings of Conference on Computational Linguistics: Posters*, pages 507–516, Mumbai, India, 2012. Conference on Computational Linguistics. (Cited on page 123.)

Heng Ji. Mining name translations from comparable corpora by creating bilingual information networks. *Proceedings of the 2nd Workshop on Building and Using Comparable Corpora : from Parallel to Non-parallel Corpora*, pages 34–37, 2009. (Cited on page 7.)

J.J. Jiang and D.W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. *Proceedings of the International Conference on Research in Computational Linguistics*, pages 19–33, 1997. (Cited on page 60.)

I T Jolliffe. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, pages 37–52, 1986. (Cited on page 93.)

Ian T. Jolliffe. *Principal Component Analysis*. Springer, 2002. (Cited on pages 26 et 34.)

Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972. (Cited on page 28.)

Leonard Kaufman and Peter J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. 1990. (Cited on page 116.)

Stefan Kaufmann. Second-order cohesion. *Computational Intelligence*, pages 511–524, 2000. (Cited on pages 14, 51, 91 et 92.)

Martin Kay. Text-translation alignment. pages 121–142, 1991. (Cited on page 10.)

Adam Kilgarriff and Gregory Grefenstette. Introduction to the special issue on the web as corpus. In *Association for Computational Linguistics*, 2003. (Cited on page 5.)

Svetla Koeva, Ivelina Stoyanova, Rositsa Dekova, Borislav Rizov, and Angel Genov. Bulgarian x-language parallel corpus. pages 23–25, 2012. (Cited on page 13.)

Tuomo Korenius, Jorma Laurikkala, and Martti Juhola. On prinicpal component analysis, cosine and euclidean measures. *Information Sciences*, 177(22):4893–4905, 2007. (Cited on pages 40 et 41.)

Sylvain Lamprier, Tassadit Amghar, Bernard Levrat, and Frederic Saubion. Using an evolving thematic clustering in a text segmentation process. *Universal Computer Science*, 14, 2008. (Cited on page 50.)

Thomas K Landauer and Susan T. Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, pages 211–240, 1997. (Cited on pages 26 et 32.)

Thomas K Landauer, Peter W. Foltz, and Darrell Laham. Introduction to latent semantic analysis. In *Discourse Processes*, 1998. (Cited on pages 33, 34 et 59.)

Claudia Leacock and Martic Chodorow. Combining local context and wordnet sense similarity for word sense identification. *WordNet, An Electronic Lexical Database*, pages 265–284, 1998. (Cited on page 60.)

Michael Lesk. Automatic sense disambiguation using machine readable dictionaries : How to tell a pine cone from an ice cream cone. *Proceedings of the SIGDOC Conference*, pages 24–26, 1986. (Cited on page 60.)

David D. Lewis. Evaluating and optmizing autonomous text classification systems. In *Proceedings of SIGIR-95, 18th ACM International Conference on Research and Development in Information Retrieval*, 1995. (Cited on page 39.)

James Lewis, Stephan Ossowski, Justin Hicks, Mounir Errami, and Harold R. Garner. Text similarity: an alternative way to search medline. *Bioinformatics (Oxford, England)*, 22(18):2298–304, 2006. (Cited on page 41.)

Dekang Lin. An information-theoretic definition of similarity. In *ICML*, pages 296–304, 1998a. (Cited on page 20.)

Dekang Lin. An information-theoretic definition of similarity. *Proceedings of the 15th International Conference on Machine Learning*, pages 296–304, 1998b. (Cited on page 60.)

Adam Lopez. Statistical machine translation. In *ACM Computing Survey*, 2008. (Cited on pages 6 et 9.)

Hatmi M, Jacquin C, Morin E, and S Meignier. Incorporating named entity recognition into the speech transcription process. In *Interspeech*, Lyon, France, 2013. (Cited on page 123.)

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, 2011. Association for Computational Linguistics. (Cited on page 34.)

Pavel Makagonov, Mikhail Alexandrov, and Alexander Gelbukh. Clustering abstracts instead of full texts. *Proceedings of the 7th International Conference on Text, Speech, Dialog (TSD), Lecture notes in Artificial Intelligence*, 3206:129–135, 2004. (Cited on pages 64 et 121.)

Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, 1999. (Cited on pages 42, 52 et 64.)

Christopher D. Manning, Prabhakar Raghava, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008. (Cited on pages 28, 29 et 31.)

Erwin Marsi and Emiel Krahmer. Annotating a parallel monolingual treebank with semantic similarity relations. In *The Sixth International Workshop on Treebanks and Linguistic Theories*, 2007. (Cited on page 8.)

Mark T. Maybury. Discourse cues for broadcast news segmentation. In *Conference on Computational Linguistics-Association for Computational Linguistics*, pages 819–822, 1998. (Cited on page 53.)

Marina Meila. Comparing clusterings - an information based distance. In *Journal of Multivariate Analysis*, volume 98(5), page 873–895, 2007. (Cited on page 68.)

Igor Melčuk. Meaning-text models: A recent trend in soviet linguistics. volume 10, pages 27–62, 1981. (Cited on page 19.)

Rada Mihalcea and Courtney Corley. Corpus-based and knowledge-based measures of text semantic similarity. In *AAAI'06*, pages 775–780, 2006. (Cited on pages 55, 58, 59 et 121.)

Andrei Mikheev. The ltg part of speech tagger. pages 50–57, 1997. (Cited on page 59.)

Jasmina Milicevic. A short guide to the meaning-text linguistic theory. In *Journal of Koralex*, pages 187–233, 2006. (Cited on page 19.)

Hajime Mochizuki, Takeo Honda, and Manabu Okumura. Text segmentation with multiple surface linguistic cues. In *Proceedings of Conference on Computational Linguistics-Association for Computational Linguistics*, pages 881–885, 1998. (Cited on page 18.)

Dragos Stefan Munteanu, Alexander Fraser, and Daniel Marcu. Improved machine translation performance via parallel sentence extraction from comparable corpora. *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 265–272, 2004. (Cited on page 7.)

Preslav Nakov, Antonia Popova, and Plamen Mateev. Weight functions impact on lsa performance. In *EuroConference on Recent Advances in Natural Language Processing*, pages 187–193, 2001. (Cited on pages 29, 33 et 122.)

Rani Nelken and Stuart M. Shieber. Towards robust context-sensitive sentence alignment for monolingual corpora. In *European Chapter of the Association for Computational Linguistics*, 2006. (Cited on pages 39, 55, 56, 58 et 121.)

Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, pages 849–856. MIT Press, 2001. (Cited on page 63.)

Hwee Tou Ng, Wei Boon Goh, and Kok Leong Low. Feature selectiouin, perceptron learning, and a usability case study for text categorization. In *Proceedings of SIGIR-07, 20th ACM International Conference on Research and Development in Information Retrieval*, pages 67–73, 1997. (Cited on page 27.)

Richard Nock and Frank Nielsen. On weighting clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28, 2006. (Cited on page 108.)

Rebecca J. Passonneau and Diane J. Litman. Intention-based segmentation: Human reliability and correlation with linguistic cues. In *Association for Computational Linguistics*, pages 148–155, 1993. (Cited on page 53.)

Jennifer Pearson. *Terms in Context - Studies in Corpus Linguistics*. John Benjamins, 1998. (Cited on page 6.)

Juan Pino and Maxine Eskenazi. An application of latent semantic analysis to word sense discrimination for words with related and unrelated meanings. In *Proceedings of the Fourth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 43–46, Boulder, Colorado, 2009. Association for Computational Linguistics. (Cited on page 32.)

David Pinto and Paolo Rosso. Kncr: A short-text narrow-domain subcorpus of medline. In *TLH 2006. Advances in Computer Science*, pages 266–269, 2006. (Cited on pages 6, 11, 112 et 121.)

David Pinto, Héctor Jiménez-Salazar, and Paolo Rosso. Clustering abstracts of scientific texts using the transition point technique. In *International Conference on Intelligent Text Processing and Computational Linguistics*, volume 3878, pages 536–546, 2006. (Cited on page 64.)

David Pinto, José-Miguel Benedí, and Paolo Rosso. Clustering narrow-domain short texts by using the kullback-leibler distance. In *International Conference on Intelligent Text Processing and Computational Linguistics*, 2007. (Cited on pages 26, 42, 64, 65, 67 et 121.)

J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993. ISBN 1-55860-238-0. (Cited on page 53.)

William M. Rand. Objective criteria for the evaluation of clustering methods. In *Journal of the American Statistical Association*, volume 66(336), page 846–850, 1971. (Cited on page 67.)

Marta Recasens, Marie-Catherine de Marneffe, and Christopher Potts. The life and death of discourse entities: Identifying singleton mentions. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 627–633, Atlanta, Georgia, 2013. Association for Computational Linguistics. (Cited on page 123.)

Roi Reichart and Ari Rappopor. The nvi clustering evaluation measure. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, page 165–173, 2009. (Cited on page 68.)

Philip Resnik. Using information content to evaluate semantic similarity. *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 448–453, 1995. (Cited on page 60.)

Jeffrey C Reynar. An automatic method of finding topic boundaries. In *Association for Computational Linguistics*, pages 331–333, 1994. (Cited on pages 47, 48 et 125.)

Jeffrey C. Reynar. Topic segmentation: Algorithms and applications. *PhD. Thesis*, 1998. (Cited on page 49.)

S.E. Robertson, S. Walker, S. Jones, M.M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. pages 109–126, 1996. (Cited on page 28.)

Andrew Rosenberg and Julia Hirschberg. Comparing clusterings - an information based distance. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, page 410–420. Association for Computational Linguistics, 2007. (Cited on page 68.)

Gerard Salton. Mathematics and information retrieval. In *Journal of Documentation*, pages 1–29, 1979. (Cited on pages 26 et 30.)

Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. In *Information Processing and Management*, pages 513–523, 1988. (Cited on page 28.)

Gerard Salton and Michael J. McGill. Introduction to modern information retrieval, 1986. (Cited on page 27.)

Gerard Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of ACM*, 18:613–620, 1975. (Cited on page 88.)

Igor Santos, Carlos Laorden, Borja Sanz, and Pablo G. Bringas. Enhanced topic-based vector space model for semantics-aware spam filtering. *Proceedings of Expert Systems With Applications*, pages 437–444, 2012. (Cited on page 31.)

Hinrich Schütze, David A. Hull, and Jan O. Pedersen. A comparison of classifiers and document representations for the routing problem. In *Annual ACM conference on Research and Development in Information Retrieval - ACM SIGIR*, pages 229–237, 1995. (Cited on page 26.)

Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34:1–47, 2002. (Cited on page 27.)

John Sinclair. Eagles preliminary recommendations on corpus typology, 1996. (Cited on page 6.)

Amit Singhal, Chris Buckley, and Mandar Mitra. Pivoted document length normalization. In *19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 21–29, 1996. (Cited on page 28.)

Helmuth Spath. *Cluster Dissection and Analysis: Theory, Fortran Programs, Examples*. Ellis Horwood, 1985. (Cited on page 62.)

Nicola Stokes, Joe Carthy, and Alan F. Smeaton. Segmenting broadcast news streams using lexical chains. In *Proceedings of 1st Starting AI Researchers Symposium*, pages 145–154, 2002. (Cited on page 47.)

George Tsatsaronis and Vicky Panagiotopoulou. A generalized vector space model for text retrieval based on semantic relatedness. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 70–78, 2009. (Cited on pages 31 et 32.)

Peter D. Turney. Mining the web for synonyms: Pmi-ir versus lsa on toefl. *Proceedings of Twelfth European Conference on Machine Learning*, pages 491–502, 2001. (Cited on page 59.)

Amos Tversky. Features of similarity. In *Psychological Review*, volume 84, pages 327–352, 1977. (Cited on page 18.)

Masao Utiyama and Hitoshi Isahara. A statistical model for domain-independent text segmentation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2001. (Cited on page 51.)

Jean Véronis and Philippe Langlais. Evaluation of parallel text alignment systems: The arcade project. pages 369–388, 2000. (Cited on page 9.)

Om Vikas, Akhil K. Meshram, Girraj Meena, and Amit Gupta. Multiple document summarization using principal component analysis incorporating semantic vector space model. *Associtaion for Computational Linguistics and Chinese Language Processing*, 13:141–156, 2008. (Cited on page 34.)

Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17:395–416, 2007. (Cited on page 63.)

William Yang Wang, Kapil Thadani, and Kathleen R. McKeown. Identifying event descriptions using co-training with online news summaries. *Proceedings of the 5th International Joint Conference on Natural Language Processing (IJCNLP 2011)*, 2011. (Cited on page 12.)

Dominic Widdows. Geometry and meaning. In *Center for the Study of Language and Information*, 2004. (Cited on page 18.)

Erik Wiener, Jan O. Pedersen, and Andreas S. Weigend. A neural network approach to topic spotting. In *Proceedings of SDAIR-95, 4th Annual Symposium on Document Analysis and Information Retrieval*, 1995. (Cited on page 27.)

Fridolin Wild, Christina Stahl, Gerald Stermsek, and Gustaf Neumann. Parameters driving effectiveness of automated essay scoring with lsa. *Proceedings of the 9th CAA*, 2005. (Cited on page 33.)

S. K.M. Wong, W. Ziarko, V. V. Raghavan, and P. C.N. Wong. On modeling of information retrieval concepts in vector spaces. *ACM Transactions on Database Systems*, 12:299–321, 1987. ISSN 0362-5915. (Cited on page 31.)

Zhibiao Wu and Martha Palmer. Verb semantics and lexical selection. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 133–138, 1994. (Cited on page 60.)

Yaakov Yaari. Segmentation of expository texts by hierarchical agglomerative clustering. *CoRR*, 1997. (Cited on page 50.)

J. P. Yamron, I. Carp, L. Gillick, S. Lowe, and P.van Mulbregt. A hidden markov model approach to text segmentation and event tracking. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 333–336, 1998. (Cited on page 54.)

Na Ye, Jingbo Zhu, Huizhen Wang, Matthew Y. Ma, and Bin Zhang. An improved model of dotplotting for text segmentation. In *Journal of Chinese Language and Computing*, pages 27–40, 2006. (Cited on page 47.)

George Kingsley Zipf. *Human Behaviour and the Principle of Least Effort: an Introduction to Human Ecology*. Addison-Wesley, 1949. (Cited on page 26.)