Using Explicit Semantic Analysis to Link in Multi-Lingual Document Collections

Final Report

Author: Lukáš Žilka
Supervisor: Petr Knoth

Knowledge Media Institute (KMi)
The Open University
Milton Keynes, United Kingdom

September 2, 2011

Contents

Acknowledgments

First of all, I want to thank to my supervisor Petr Knoth for his guidance along the whole way from the time I came to KMi until the end when my work here is finished. Not only has he always been very helpful and supporting, which kept motivating me towards the completion of my project, but also he gave me priceless advices and introduced me to the environment of KMi in Milton Keynes which had been completely unknown to me. Many thanks also to my parents and my sister who have never ceased supporting me in all possible ways and to whom my great gratitude belongs. Then, I want to say a BIG thank you to all people at KMi that helped me progress with my work and made the entire experience pleasant, unforgivable and without whom I would have much harder time to finish my work. You are all amazing and he is lucky who comes to KMi after me: Maria, Petr, Magda, Vojta, Jacek, Hassan, Miriam, Philip, Nico, Carlo, Loukas, Matt, Carlos, Andriy, Claudia, Ning, Chengua, Martin, Anna-Lisa, Vanessa, Sofia, Fouad, Enrico, Zdenek, Thomas, Alba, Massimiliano, Suzanne, Keerthi, Fridolin, Bassem, Ortenz, Lewis, Robbie, Joe, Alan, Mark, Ben, Shyamalla, Peter, John. I will never forget the fun we had at table football games, squash sessions, night shifts, cooking, movie nights, game gatherings, soccer tournament, paintball, bowling, birthday events and the amazing trips with many of you. Also, many thanks to the EU Erasmus program and Brno University of Technology that supplied the financial resources for my stay at KMi, to KMi that provided the excellent base for my work, to Martin Pelikán and Steadynet s.r.o, who provided the much needed computational power for my experiments when it was needed, and to Pavel Smrž from FIT, Brno University of Technology, who empowered me with access to additional computer equipment.

Abstract

Keeping links in quickly growing document collections up-to-date is problematic, which is exacerbated by their multi-linguality. We utilize Explicit Semantic Analysis to help identify relevant documents and links across languages without machine translation. We designed and implemented several approaches as a part of our link discovery system. Evaluation was conducted on Chinese, Czech, English and Spanish Wikipedia. Also, we discuss the evaluation methodology for such systems and assess the agreement between links on different versions of Wikipedia. In addition, we evaluate properties of Explicit Semantic Analysis which are important for its practical use.

Assignment

1. Familiarize yourself with existing approaches to automatic link generation.
2. Study the approach cross-language discovery of links and the evaluation approaches on the NTCIR CrossLingual Link Discovery Task (http://ntcir.nii.ac.jp/CrossLink/)
3. Discuss with the team at KMi, OU possible hypotheses for monolingual and cross-lingual link discovery. Design in collaboration with KMi a variety of new approaches to be tested.
4. Implement the designed methods by extending the existing SemRel Analyzer system.
5. Test the methods on two use-cases: a) The NTCIR tasks b) The dataset of research papers acquired through the CORE project.
6. Discuss the results, their advantages and drawbacks, compare to other existing similar solutions and describe possible extensions.
7. Collaborate on a conference submission (conference to be selected by KMi, OU).

Chapter 1

Introduction

Cross-referencing documents is an essential part of organising textual information, as with the expanding amount of the textual information produced every day, finding the relevant information is getting increasingly difficult. However, keeping links in large, quickly growing document collections up-to-date is due to the number of possible connections problematic, and in addition, this tends to be subjective and time demanding when done manually. In multilingual collections, such as Wikipedia, collections of scientific articles, or articles on the web, interlinking semantically related information in a timely manner becomes even more challenging, and nearly impossible for the incredible amount of information that a person would have to be familiar with to produce high quality links.

There is currently no software that could facilitate the automatic cross-lingual link discovery. Therefore, here we aim to design and implement a system that helps to keep up-to-date cross-lingual links in a document collection by automatic cross-referencing documents and automatic context-link discovery.

In this work we describe the design and implementation of such cross-lingual linking system which given a document written in English and a collection of documents written in another language, finds relevant documents and context links. We devised several approaches, all of them based on the Explicit Semantic Analysis [5], Geva's Algorithm [6] and Itakura's Algorithm [12]. Although some of them are applicable to general document collections and other exploit specific properties of some document collections, due to our inability to find another suitable [1] document collection, all of the approaches were evaluated on Wikipedia.

First chapter describes the current state of the link discovery field, and other relevant areas. In second chapter the design of our cross-lingual link discovery system is described. Third chapter describes the implementation. Fourth chapter is about evaluation and other conducted experiments, and the last one gives a summary along with future research directions.

1.1 Motivation

This work has been largely motivated by participation of KMi in the NTCIR-9:CrossLink competition. The competition asks the participants to create a system that is able to find contextual links in the pages extracted from the English Wikipedia pointing to the Chinese, Japanese and Korean (CJK) ones. The idea behind this task lies in the lack or

[1] That means a collection that is already interlinked and is multilingual.

quality of information about some cultural phenomenas in a particular version of Wikipedia. A bilingual reader could thus benefit from the cross-lingual links which would take them from a rich page which is high quality in English Wikipedia to a high quality pages in CJK Wikipedias which in English are stubs or non-existent.

1.2 Related Work

To the best of our knowledge no work has been yet published on the topic of cross-lingual link discovery, but the work that we mainly built on are from the field of information retrieval and its sub-field of mono-lingual link discovery.

1.2.1 Linking Algorithms

Among the others, two successful approaches towards link detection were presented at the INEX 2007 Link-the-Wiki track – Itakura's Algorithm and Geva's Algorithm. Both of them are based on the knowledge of the link structure in the document collection (particularly Wikipedia). The basic versions of the algorithms were also used in [13] and [18] in their mono-lingual linking systems. We adopted them in our system as well, by adjusting them to work cross-lingually.

Itakura's Algorithm builds an index of links existing in the document collection (link text; target), and then uses this index to find links on the linked page [12].

Geva's Algorithm builds an index of page titles of the document collection which then uses to find links on the linked page [6].

Wikisearching and Wikilinking In [13] the authors implement a slightly modified version of the aforementioned Itakura's and Geva's algorithms. Their modification mainly lies in normalisation of the text. As a result, improving the overall results of those two algorithms.

Learning to Link with Wikipedia The authors of [18] has came up with yet another solution to linking documents in Wikipedia. They describe how they can achieve high recall and precision values, using the machine learning approach. They first build two classifiers – disambiguator and link detector. The disambiguator is used for picking a sense of a word in a document, and then they use the link detector for deciding whether a given keyword should be a link or not.

Although this approach is mono-lingual, in our work, we adopted the link decision part while we tried to address the disambiguation issue by applying the Explicit Semantic Analysis. Also, we added the cross-lingual step.

1.2.2 An Experimental Comparison of Explicit Semantic Analysis Implementations for Cross-Language Retrieval

The authors of [24] aimed at finding the best parameters of the Explicit Semantic Analysis method. They systematically evaluated different behaviour of the Cross-lingual Explicit Semantic Analysis when applied to the Information Retrieval problem. In our systems we used the parameters that had the optimal results in their experiments.

1.3 Cross-lingual Link Discovery System Specification

Given an input document in one language and a target document collection in another language, our goal is to produce link suggestions by finding specific places in the input document (anchors) which point to specific documents in the target collection. A correct link suggestion is such that either points to a document in the target collection which is, judged by people, relevant to the input document and develops the concepts introduced in it, or points to a document in the target collection which describes or gives insight of particular concept from the input document.

In our work, the quality of the system's link suggestions will be automatically assessed by running the system on Wikipedia, which is a special case of a multi-lingual document collection.

The limitations on the language of the input document and of the target collection are imposed by the methods that we aim to use. The core method that we use needs a parallel multi-lingual background document collection in each of the supported languages. So the limitation is given by the availability of such parallel document collections in different languages [2]. Structural limitations on the target document collection vary over different approaches, from no limitation, over the need of the collection to be already interlinked, to the requirement that the collection has the conceptual character.

[2]We use Wikipedia as the background in our methods, therefore languages which have a reasonable Wikipedia article base (100k+ articles), which nowadays is the majority of languages, can be supported.

Chapter 2

Background

The cross-lingual link discovery system is based on knowledge from the area of information retrieval, particularly its subareas dealing with semantic similarity computation and link discovery methods. In this chapter the relevant sub-areas are introduced in detail that is necessary for understanding the approaches described later in this work.

2.1 Link Discovery

Link Discovery is a field that deals with automatic finding of contextual links in documents. The links point from a given document into a document collection. A special case of such document collection is for example Wikipedia. Because Wikipedia is a general and easily available document collection, many current link discovery methods were built especially for inter-linking within or linking from an external document into Wikipedia itself. Although, there are also attempts to build universal linking systems which are able to inter-link arbitrary document collections.

The current approaches to link discovery can be divided into the following groups:

(1) *link-based* approaches discover new links by exploiting an existing link graph [12, 13, 16].

(2) *semi-structured* approaches try to discover new links using semi-structured information, such as the anchor texts or document titles [6, 4, 8].

(3) *purely content-based* approaches use as an input plain text only. They typically discover related resources by calculating semantic similarity based on document vectors [1, 9, 25, 27, 11].

(4) *combined* approaches combine more methods. For example [16] use the semi-structured approach in the first step, to identify the candidates, and then they use the link-based approach to rank the candidates.

We have tried to apply Explicit Semantic Analysis in several ways, so our systems fall in each of the aforementioned categories.

2.2 Auxiliary Methods

Cosine similarity and TF-IDF are two methods from information retrieval that are used in Explicit Semantic Analysis and are needed for understanding its principles.

2.2.1 TF-IDF

TF-IDF is a standard measure in the field of Information Retrieval for evaluating the importance of a term in a document collection [22]. Its two components are:

Term Frequency (TF) says that a term is important if it is repeated a lot in the given document.

IDF says that a term is important if its occurrence over the whole document collection is sparse.

Let D be a document collection, let $d \in D$ be a document from that collection, let t be a term that figures in the document collection, and let $tf_{t,d}$ be a number of occurrences of the term t in the document d, the formula for the TF-IDF computation is the following:

$$\text{tf(t, d)} = log(tf_{t,d}) \tag{2.1}$$

$$\text{idf(t)} = log(\frac{|D|}{|\{d : tf_{t,d} > 0\}|}) \tag{2.2}$$

$$\text{tf.idf(t, d)} = tf(t, d) \cdot idf(t) \tag{2.3}$$

2.2.2 Cosine Similarity

Cosine similarity is a standard measure for computing similarity of two vectors. It basically measures the angle between the vectors as a real number and lies between 1 (if the vectors are the same) and 0 (if the vectors are orthogonal).

$$cos(A, B) = \frac{A \cdot B}{\|A\|\|B\|} \tag{2.4}$$

2.3 Explicit Semantic Analysis

Explicit Semantic Analysis (ESA) is a method for computing semantic relatedness of two texts devised by [5]. It aims to represent any given text (input text) by a vector of weighed and explicit concepts (ESA vector), and compute relatedness as the similarity of those vectors. Each dimension of the ESA vector represents a concept, and the value of each dimension of the ESA vector represents the association strength of the input text with that concept. So the ESA vector as a whole represents how much of which concept is contained in the input text. The concepts are given by the background collection (Wikipedia in most cases, so in case of the English Wikipedia, there are about 3.5 million concepts), and give, so far, the best results in computing semantic similarity of texts than anything else (LDA, LSA, bag-of-words) [5].

The idea behind Explicit Semantic Analysis is illustrated in Figure 2.1 and the schema of the whole process of figuring out the semantic relatedness is shown in Figure 2.2.

2.3.1 Explicit Semantic Analysis Background Inverted Index

Explicit Semantic Analysis needs for its operation an inverted index of the background collection weighed by TF-IDF the terms in the indexed documents. The index is needed for a part of ESA called Semantic Interpreter which with help of the index maps documents into ESA concept space.

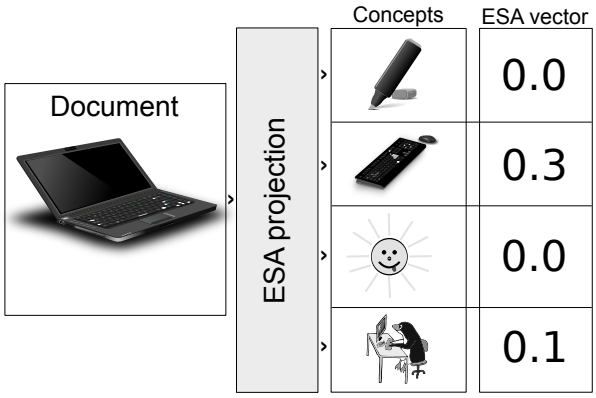

Figure 2.1: The idea of Explicit Semantic Analysis. Projection of a document into the explicit concept space. Concept space comprises "pen", "keyboard and mouse", "sun", "office". Values in the ESA vector are the association strengths of the document with individual concepts.

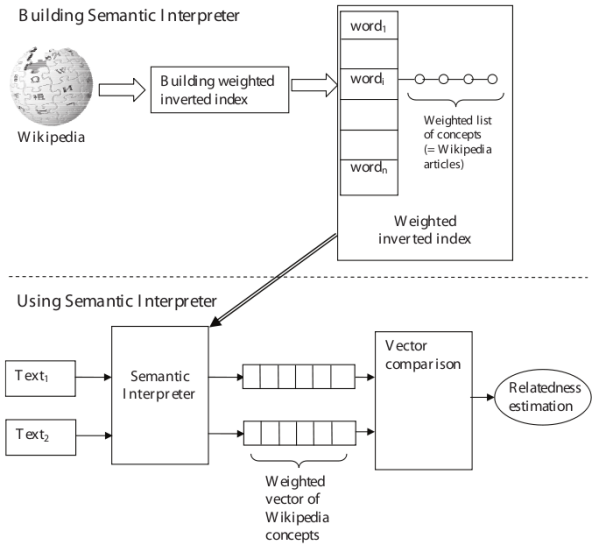

Figure 2.2: Schema of a system providing semantic relatedness computation based on Explicit Semantic Analysis. Figure reprinted from [5].

The process of building the inverted index is influenced by several variables such as choice of the metric for determining term's importance in text or choice of the document collection. Both are discussed in Subsection 2.3.3.

2.3.2 Formal Definition

Explicit Semantic Analysis was formally defined in [5] as follows. Let $T = W_i$ be input text, and let $\langle v_i \rangle$ be its TF-IDF vector, where v_i is the weight of word w_i. Let $\langle k_j \rangle$ be an inverted index entry for word w_i, where k_j quantifies the strength of association of word w_i with Wikipedia concept $c_j, \{cj \in c_1, ..., c_N\}$ (where N is the total number of Wikipedia

concepts). Then, the ESA vector V for text T is a vector of length N, in which the weight of each concept c_j is defined as $\sum_{w \in T} v_i \cdot k_j$. Entries of this vector reflect the relevance of the corresponding concepts to text T.

2.3.3 Variables

There are several places in the ESA method that can be tuned and influence the performance:

Association Function computes how much the given concept represents the input text and gives results that lie in the interval $< 0; 1 >$. In [24] it was experimentally measured that that TF.IDF* [1] yields the best results. Alternatives are: TF.IDF, TF, BM25, Cosine, and possibly others.

Dimension Projection Function reduces the number of dimensions of a term in the background index. The reduction here is important as it would be, basically, computationally impossible to work with ESA on the current hardware. ESA uses the background index for determining the association strength between the input text and explicit concept, effectively building the ESA vector for the input text. The best results are yielded when the function crops the vector after the 10k best dimensions. Other considered alternatives were the sliding window (the original dimension projection function, proposed by the authors of ESA), absolute threshold, and relative threshold.

Vector Similarity Measure computes similarity of two ESA vectors in order to determine the semantic relatedness of the documents which the ESA vectors represent. It was experimentally discovered that Cosine similarity measure yields the best results. Alternatives are: KL-Divergence, LM, TF.IDF (transferred to the bag-of-articles).

(Number of) Concepts It was shown in [2] that the choice of the background corpus is important, but it is not known yet what is the ideal background corpus as a random background corpus achieved similar results as other corpora which were meaningful (Reuters, Wikipedia).

2.3.4 Cross-lingual ESA

Cross-lingual ESA is an exploitation of ESA to cross the language barrier. It makes use of the two unique properties of Wikipedia – links between an article and its counterparts in other languages, and occurrence of articles on the same topic in other languages, and has been introduced by [23]. If we take the Wikipedia articles as concepts for ESA, then, using Wikipedia, we can build an ESA background for mapping an arbitrary document written in any language that has a sufficient article-base in Wikipedia into the specific language version's Wikipedia concept space [2]. So if we know the mapping between corresponding concepts in different languages we can compare the ESA vectors across languages, and as a result, we can use the vectors as a cross-lingual carrier of document's semantics.

[1] The star denotes that in TF.IDF formula the word-occurrence count term for the query document is omitted.

[2] This is exactly what the original version of ESA does. Uses Wikipedia articles as concepts, and then maps an arbitrary text into the space created by the articles.

Formally, we can define a finite set of universal concepts C_U which is an intersection of Wikipedia articles from language 1 C_{L_1} and Wikipedia concepts from language 2 C_{L_2}. In other words, it is a set of concepts that are common to both language versions of Wikipedia:

$$C_U = C_{L_1} \cap C_{L_2}$$

Then, an ESA vector of any document is its projection into the concept space $\mathbb{R}^{C_U}$. That is the same for both languages (L_1 and L_2) and since the vectors are in the same space it is possible to apply standard vector similarity measures on them, to compare semantic similarity of documents in different languages.

2.4 Link Discovery Task Evaluation

2.4.1 Metrics: Precision and Recall

Precision and recall are standard measures heavily used in the field of information retrieval to measure performance of information retrieval systems [21]. The idea is that each document is marked either as relevant or non-relevant and the total numbers of such documents are then used in the evaluation metrics along with the numbers from a ground-truth. Precision, which in a way measures the quality of produced results, is the ratio of the number of the relevant documents to the number of retrieved documents. Recall, which in a way measures the completeness of the results produced by the system, is the ratio of the number of relevant documents to the number of documents that are in the ground-truth (which is the number of the correct documents).

Let D_{GT} be a finite set of all relevant documents from the ground-truth, let D_{RR} be a finite set of retrieved relevant documents, and let D_{RN} be a finite set of retrieved non-relevant documents.

$$\text{Precision} = \frac{|D_{RR}|}{|D_{RR}| + |D_{RN}|} \tag{2.5}$$

$$\text{Recall} = \frac{|D_{RR}|}{|D_{GT}|} \tag{2.6}$$

Both metrics can be computed at different places of the system (e.g. after different number of retrieved documents), and produce so called Precision-Recall graphs which are used for determining the desired configuration properties of the system.

Mean Average Precision

Mean Average Precision is a mean of average precisions over all documents in a testing set D [14]. It can be computed as follows:

$$\text{MAP} = \frac{\sum_{d=1}^{D} \text{AverageP(d)}}{|D|} \tag{2.7}$$

2.4.2 Data: Wikipedia

We use Wikipedia as a source of evaluation data as well as the background for Explicit Semantic Analysis. Wikipedia helps us evaluate the link suggestions of our systems. The

systems are let to reproduce the links from Wikipedia which are subsequently compared with the currently existing ones. Thus, we can assess the systems' performance.

There are several properties of Wikipedia that we exploit for evaluation. Wikipedia is well interlinked. Each of the articles is connected with the rest, and only 8 per cent of them do not belong to the biggest strongly connected component of the Wikipedia's link graph [3]. Also, its another strong feature is multi-linguality. Articles about basic phenomenas exist across languages and are explicitly grouped together. Both features give us an opportunity to build a ground truth to test our systems.

2.4.3 Ground truth

Ground truth is generally a set of results for a task that are thought to be the perfect solution. It serves as a reference for comparing results of systems that try to solve this task.

2.4.4 Mono-lingual Link Discovery Task Evaluation

INEX's Link-the-Wiki Track evaluates the linking task by creating a ground truth that consists of the links contained on the page that is given to the linking system. In other words, the input page is taken from Wikipedia, is orphaned (= all links are removed from the page) and the links are stored as the ground-truth. Then, the links suggested by the mono-lingual link discovery system are taken and compared to the ground-truth, with precision and recall metrics as a result.

2.4.5 Cross-lingual Link Discovery Task Evaluation

INEX's CrossLink Track measures precision and recall of the system. Two ways for determining relevance/non-relevance are set:

Wikipedia based (automatic) evaluation constructs the ground-truth from the links on the input Wikipedia page and its counterpart in the other language. From the input page it takes only links whose targets have counterparts in the other language's Wikipedia. From the input page's counterpart, all links are in the ground-truth. Then, when a page annotated by the cross-lingual link discovery system is evaluated, only links that are in the ground-truth for the particular page are marked as relevant, others are marked as non-relevant. This evaluation does not deal with the position of the link in the Wikipedia page.

Human based (manual) evaluation presents the pages, were annotated by the cross-lingual discovery system with links, to human annotators. People then, mark each link as either relevant or non-relevant.

Precision and recall are then measured the standard way, as described in Subsection 2.4.1. Also other metrics such as MAP (Section 2.4.1) can be computed.

Chapter 3

Cross-lingual Linking Discovery: Design

In this chapter we define the cross-lingual link discovery task, define the model of our linking system, describe the methods used in our experiments, and also discuss the evaluation model.

The main issues that a linking system needs to address are the discovery of potential link candidates, and determining the relatedness of the candidate links (or link validity) which goes hand in hand with the completeness of the resulting link suggestions. The two latter mentioned pose a problem for us as well as for the computers. Relatedness is one that is easier for us to solve, but we struggle with completeness; as we understand text and can tell with high confidence if two texts are relevant, but the task is hard for us in terms of completeness, as we do not have every potential document in our head so we often miss a lot of relevant links. Whereas, computers so far do not understand the text, thus evaluating relatedness of an article is an issue, but can quite easily store and completely search document collections for some particular information so that every possibly relevant document is considered.

3.1 Terminology

In this chapter the following terms are used to describe our cross-lingual link discovery system and its components:

Natural language is any spoken/written human language (such as English, Spanish, Czech).

Document is a text written in one of the natural languages (such as a Wikipedia article about Kangaroos).

Ranked Document is a tuple of a document and a real number.

Document collection is a finite set of documents all of which are written in the same language (such as English Wikipedia).

Link is a connection between two documents (such as a link between the document about Australia and the document about Kangaroos).

Concept is a document from the background collection of the Explicit Semantic Analysis method (such as an article about Physics from the English Wikipedia).

3.2 Cross-lingual Link Discovery Task Definition

Let L_1, L_2 be natural languages. Let d_{in} be an input document written in L_1, and let D be a document collection written in L_2. The task of cross-lingual linking lies in finding a subset of documents from D, which, judged by people, are in terms of content of d_{in} either relevant, or explain or develop concepts introduced by d_1.

3.3 Cross-lingual Link Discovery System

The system addresses the cross-lingual link discovery task defined in the previous section and is composed of the following components, which are discussed in detail in the following section:

1. Link Discovery. In this stage the system tries to identify all possible targets for links from the input document. One of the following approaches is used:
   - CL-ESADirect
   - CL-ESA2Links (or ESA Link Base)
   - CL-ESA2Similar
   - CL-ESA2ESA
   - Terminology
2. Link Placement. Here, the system filters the discovered links by trying to fit them into the input document. If the link cannot be placed, it is thrown away.
3. Link Classification. This phase serves as the final filter where the final link candidates are picked.

3.4 Auxiliary Definitions

In this section, auxiliary formal devices are introduced to facilitate exact, easier and more comprehensive description of the link discovery mechanisms.

3.4.1 Links in a Document Collection

Let L be a natural language. Let D be a document collection written in L. Let λ be a function, that assigns each document $d \in D$ a finite set of documents to which d links to:

$$\lambda : D \to 2^D \tag{3.1}$$

$$\lambda(d) \subseteq D \tag{3.2}$$

Let Π_D be a set of all existing links in the document collection D:

$$\Pi_D = \bigcup_{d}^{D} \{(d, x) : x \in \lambda(d)\} \tag{3.3}$$

3.4.2 Cross-lingual Mapping

Let L_1, L_2 be natural languages. Let D_1, D_2 be document collections written in L_1 and L_2, respectively. Let $\rho_{1\to2}$ be a mapping function, that assigns a document from D_1 its counterpart(s) [1] from D_2:

$$\rho_{D_1\to D_2} : D_1 \to 2^{D_2} \tag{3.4}$$

$$\rho_{D_1\to D_2}(d_1) \subseteq D_2 \tag{3.5}$$

This represents the cross-lingual mapping for pairs of documents. One document is the counterpart of the other one in the other document collection. The documents from both collections which are linked together by $\rho_{D_1\to D_2}$ should, therefore, be identical in terms of semantics of content (i.e. parallel corpora, or Wikipedia pages of the same topic, just in different languages). E.g.:

$$\rho_{W_{EN}\to W_{FR}}(\text{English Channel}) = \{\text{Manche}\}$$

3.4.3 Universal Concepts

In this chapter, the term universal concepts refers to the set of concepts common for both background collections used for Cross-lingual Explicit Semantic Analysis, as defined in Subsection 2.3.4.

3.4.4 Explicit Semantic Analysis

Explicit Semantic Analysis is introduced in Section 2.3 and here we set the formal denotation that will be used. Let L be a natural language, and D a document collection written in L. Let D_B be a background collection of documents written in L. Let C_U be a finite set of universal concepts. Let ϵ_L be a function, that projects each document $d \in D$ into a space of universal concepts C_U:

$$\epsilon_L : D \to \mathbb{R}^{|C_U|} \tag{3.6}$$

$$\epsilon_L(d) \in \mathbb{R}^{|C_U|} \tag{3.7}$$

Let γ_{D_B} be a partial function, mapping the universal concepts C_U onto the document collection D_B:

$$\gamma_{D_B} : C_U \to D_B \tag{3.8}$$

$$\gamma_{D_B}(c \in C_U) = d \in D_B \tag{3.9}$$

3.4.5 Vector Similarity Measure

Let L be a natural language, and D a document collection written in L. Let C_U be a finite set of universal concepts. Let τ be a function, that assigns to each pair of vectors $v_1 \in \mathbb{R}^{|C_U|}$ and $v_2 \in \mathbb{R}^{|C_U|}$) a relatedness measure $x \in \mathbb{R}$:

$$\tau : \mathbb{R}^{|C_U|} \times \mathbb{R}^{|C_U|} \to \mathbb{R} \tag{3.10}$$

$$\tau(v_1, v_2) \in \mathbb{R} \tag{3.11}$$

This represents a general model for vector similarity measures, such as Cosine similarity introduced in Subsection 2.2.2.

[1] Note that the cross-lingual mapping is not bijection, but rather an arbitrary relation.

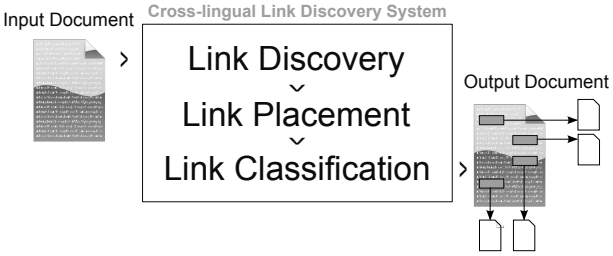

Figure 3.1: Overview schema of the Cross-lingual Link Discovery System.

3.4.6 Related Documents Discovery

Let L_1, L_2 be natural languages. Let d_{in} be an input document written in language L_1. Let D be a finite set of documents written in language L_2. Let $\Omega_k^{D,d_{in}}$ be a finite set of ranked documents in language L_2:

$$\Omega_k^{D,d_{in}} = \{(d, x) : d \in D, d_{in}^* = \epsilon_{L_1}(d_{in}), x \in \mathbb{R}, x = \tau(\epsilon_{L_2}(d), d_{in}^*)\} \tag{3.12}$$

This is a sub-task of the link discovery, later used in the link discovery methods. The set $\Omega_k^{D,d_{in}}$ is produced by mapping the input document d_{in} into the universal concept space $\mathbb{R}^{C_U}$ by ESA: $d_{in}^* = \epsilon_{L_1}(d_1)$, and measuring the similarity of the resulting ESA vector d_{in}^* with ESA vectors $d^* \in \{\epsilon_{L_2}(d) : d \in D\}$ for each document of the document collection D by the similarity measure function τ. Then, ranking of each document denotes the relatedness of the document to d_{in}. In other words, it utilizes ESA to search for semantically relevant documents written in one language with a document written in another language, effectively helping to cross the language barrier. In addition, finding a document that is semantically similar to the linked document is useful as it is very likely either a relevant link for the linked page d_{in} (as it talks about a relevant topic), or, it, already being a part of the document collection D_2, can help us search for the relevant links in the document collection.

In the following text $\Omega_k^{d,D}$, where k, d and D are parameters, denotes the set of Ω containing only the best k ranked documents from the document collection D for the document d. Sometimes, when talking generally about a set Ω the indexes are left out for the sake of clarity. Ω is always understood to have the following structure:

$$\Omega = \{(d_{o_1}, x_1), ..., (d_{o|D_2|}, x_{|D_2|})\}; \forall k, l \in \mathbb{N} : x_k < x_l \Rightarrow k > l \tag{3.13}$$

$$\tag{3.14}$$

3.5 Components of the Cross-Lingual Link Discovery System

As an input, our cross-lingual linking system takes an input document in one language and a target collection of documents in another language, and as a result produces the link suggestions. It uses a collection of documents for each of the two languages as a background for the particular language version of the Explicit Semantic Analysis. Bellow, the model of the components of the linking system is formally defined and described. Then, different compositions of the components into the cross-lingual link discovery system are discussed.

An overview schema of the link discoveery system is depicted on Figure 3.1.

3.5.1 Link Discovery

Link discovery aims to produce a short-listed candidate list of documents for an input document. We devised a number of methods which are described in this section. They have parameters which are in brackets next to their names.

Let L_1, L_2 be natural languages. Let d_{in} be an input document written in language L_1. Let D_1, and D_2 be finite sets of documents written in languages L_1, L_2, respectively. Let Ω be a finite set of documents in language L_2 ranked according to their semantic relatedness to an input document d_{in}, as in Equation 3.12. Let Y be the list of link suggestions (= discovered links) for d_{in}.

CL-ESADirect(N) This method is very straightforward and only takes the set of ranked documents Ω, picks the most semantically relevant N of them and declares them to be the link suggestions Y:

$$Y = \{d_{o_1}, ..., d_{o_N}\} \tag{3.15}$$

The presumption of this method is that a good link for the input document is such that links to a semantically similar document. So this method only uses the ranked list of similar documents in the target language Ω as the resulting list of target links Y.

CL-ESA2Links(N, k) (also called ESA Link Base in the following text) This method takes the links which already exist on the first k most semantically related pages in Ω and creates a finite set of documents D_λ out of them.

$$D_\lambda = \bigcup_{i}^{k} \{d \in D_2 : (d_{o_i}, d) \in \Pi_{D_2}\} \tag{3.16}$$

The document collection D_λ is, again, ranked according to the semantic relatedness of its documents with the linked document d_{in}, producing a set of ranked documents Ω^*, and the most semantically relevant documents from this ranked list is returned as the resulting list of the link suggestions:

$$\Omega^* = \{(d, x) : d \in D_\lambda, d^*_{in} = \epsilon_{L_1}(d_{in}), x \in \mathbb{R}, x = \tau(\epsilon_{L_2}(d), d^*_{in})\} \tag{3.17}$$

The set Ω^* is then denoted as follows, from the most relevant to the least relevant:

$$\Omega^* = \{(d_{o_1}, x_1), ..., (d_{o|D_2|}, x_{|D_2|})\}, \forall k, l \in \mathbb{N} : x_k < x_l \tag{3.18}$$

Which results in the following list of link suggestions:

$$Y = \{d_{o_1}, ..., d_{o_N}\} \tag{3.19}$$

This method requires the knowledge of the link structure in the target collection D_2. It expects that the input document d_{in} is being linked to an already interlinked collection D_2, link-structure of which this method exploits. The link targets Y are extracted from the links found on the most similar N documents from D_2 and then ranked according to their semantic similarity to the source document. This list is then used as a collection of targets.

CL-ESA2Similar(k, N) This method takes the first k most semantically related pages $S \subseteq \Omega; |S| = k$ and sums up their universal concept space vectors to compute an average universal concept space vector d^*_α:

$$d^*_\alpha = \sum_{(d,x) \in S} \epsilon_{L_2}(d)$$

Then it produces the resulting ranked list of suggested links Ω_α, based on this vector. It relies on the concept mapping γ_{D_2}, between the universal concept set C to and the document collection D_2. We denote a selection of a dimension $c \in C_U$ from a vector d^*_α as: $d^*_\alpha[c]$

$$\Omega_\alpha = \{(d, x) : d = \gamma_{D_2}(c), c \in C_U, x = d^*_\alpha[c]\} \tag{3.20}$$

The set Ω_α is denoted as follows (from the most relevant to the least relevant):

$$\Omega_\alpha = \{(d_{o_1}, x_1), ..., (d_{o|D_2|}, x_{|D_2|})\}, \forall k, l \in \mathbb{N} : x_k < x_l \tag{3.21}$$

Then, the set of suggested links Y is:

$$Y = \{d_{o_1}, ..., d_{o_N}\} \tag{3.22}$$

This method takes the average ESA vector and looks at its dimensions. The values of its dimensions represent by a real number how much is the document, which the ESA vector is for, similar to a given universal concept which the given dimension represents. The universal concept themselves in our setting are Wikipedia topics, mappable into our document collection D_2. We therefore exploit this mapping and suggest the best dimensions of this average to be the links.

CL-ESA2ESA(k) This method uses the d^*_α vector from the previous method (CL-ESA2Similar) and uses it as an ESA vector for getting a set Ω^* of ranked list of documents [2]

$$\Omega^* = \{(d, x) : d \in D_2, x \in \mathbb{R}, x = \tau(\epsilon_{L_2}(d), d^*_\alpha)\} \tag{3.23}$$

If the set Ω^* is then denoted as follows, from the most relevant to the least relevant:

$$\Omega^* = \{(d_{o_1}, x_1), ..., (d_{o|D_2|}, x_{|D_2|})\}, \forall k, l \in \mathbb{N} : x_k < x_l \tag{3.24}$$

Then, the set of suggested links Y is:

$$Y = \{d_{o_1}, ..., d_{o_N}\} \tag{3.25}$$

In this method, the top k most semantically similar documents are retrieved. Then, their ESA vectors are averaged and the resulting average vector is used to search for similar documents again. By this, this method aims to find an ESA vector of so-called average document which smooths out the semantic specificness of individual documents. The ranked list of the search for an average document is then used as the resulting list of target links.

[2] Note that d^*_α is basically an ESA vector of an average document of the most semantically similar documents in the target collection. So Ω^* is a ranked list of the most semantically similar documents to this non-existent average document.

Terminology Vocabulary This method exploits the captions of the documents to come up with link suggestions. Basically, all documents in the document collection D_2 which have cross-lingual mapping to the original collection, and therefore a known title in L_1, are suggested as link targets:

$$Y = D_2 \tag{3.26}$$

It is up to the Link Placement component of the cross-lingual link discovery system to find the useful ones.

3.5.2 Link Placement

Link Placement is an optional part of the cross-lingual link discovery system which aims to place the discovered links in the input document. It is based on the Document Vocabulary. Document Vocabulary is a structure that pairs each document with a short text that describes it (e.g. caption or link text) and can be thought of as a dictionary. This essentially makes our Link Placement method mostly usable only when a caption or label for the possible link target documents is known. Algorithm 1 locates link source in the document.

In Algorithm 1 several auxiliary functions are used. Function `tokenize(document)` splits the document into tokens and stems them. Function `in_vocabulary` checks whether a given string parameter can be mapped on an item from the dictionary, and `load_vocabulary` loads the corresponding items from the dictionary.

Algorithm 1 Link Candidate Suggestion

```
1:  procedure locate_links(doc):
2:  tokens ← tokenize(doc);
3:  candidates ← {};
4:  for t ∈ tokens do
5:     search_result ← {};
6:     search_list ← { tokens[t] };
7:     while in_vocabulary(search_list) do
8:        search_result ← search_result ∪ load_vocabulary(search_list);
9:        search_list ← search_list ∪ { tokens[t++] };
10:    end while;
11:    candidates ← candidates ∪ search_result;
12: end for;
13: return candidates;
```

3.5.3 Link Classification

The Link Discovery and Link Placement methods described above yield a great number of links that need to be further processed and filtered. For this purpose our approach uses machine learning. To facilitate this the links need to be annotated with features. We used the following features (occurrence, generality and link frequency were inspired by [18]):

ESA similarity is a real number between 0 and 1, that express how much similar given terms are. As features we included three different similarities:

- of the link text to the target document

- of the link text to the target document title
- of the input document to the target document

Generality is a measure expressing how general a given topic is. It is a natural number between 0 and 16.

Link frequency is a measure expressing how many times a particular keyword occurs as a link in the whole document collection.

Occurrence of the link text in the input document is a relative measure of the first, last, current occurrence of the link text in the input document, and the difference between its first and last occurrence.

3.6 ESA-based Document Retrieval

Our cross-lingual link discovery methods need to find semantically relevant/similar documents in huge document collections. Though, without a more sophisticated mechanism this is impossible to achieve (in terms of reasonable computation time). To accommodate this need a document retrieval engine was designed. We aimed to design a method that takes a document as an input and retrieves a number of documents ranked according to their semantic similarity with the input document. This is based on using the ESA vectors of the query document, the ESA vectors of the documents in the searched document collection, and cosine similarity function.

Even though this was designed specifically for our case of searching the ESA vectors database, our approach is generally applicable to searching in huge vector databases. There are two parts of our ESA-based document retrieval system – subset picking and searching.

Subset Picking The core of our approach lies in selecting a relevant subset of vectors which will be compared. When two vectors are being scored by the scoring function, only the common dimensions are considered. Therefore, when two vectors do not share a common dimension they do not need to be even evaluated (as the result of such comparison is the same as that of a comparison of null vectors).

Search When a search is invoked, the engine builds from the whole document collection the subset of documents described above. In the subset only documents, having at least one common dimension with the query ESA vector are included. Semantic similarity is then computed between the query vector and each ESA-vector of the documents in this subset. The best scoring vectors are then returned as a result.

This way, in practice, the similarity computation is performed in average on only about 5 % [3] documents from the whole document collection, substantially reducing the computational time.

[3] This of course depends on the characteristics of the document collection, background collection for ESA, and also on the query vector. Together they determine the size of the subset of documents that need to be considered for potential similarity.

3.6.1 Picking ESA Dimensions

Due to the fact that the ESA vector has usually 4 million dimensions, majority of which are insignificant, it is desirable to consider only certain number of them to save space and computational power. Experimentally, we found out that a reasonable compromise between information loss and computational expensiveness is to keep the first 100 dimensions of an ESA vector.

Chapter 4

Cross-lingual Link Discovery: Implementation

Building a cross-lingual link discovery system is a complex task that consists of putting many modules and parts together to collaborate in order to accomplish the given goal. The design of the individual parts of our system has been defined in Chapter 3. Existing implementations were explored and partly reused. Overall, the system's implementation is new and based on the principles introduced in the aforementioned chapter.

Data Data are expected to be the MediaWiki format stored in a standard MediaWiki database layout. Any information about the link structure that is present in the document collection is in form of wiki-markup in the text of individual articles.

Database Throughout the development MySQL has always been used as the database engine, mainly for the reason that speed was much more crucial than any advanced database features. MyISAM backend was chosen as the best option considering the speed and storage overhead. Though, the implementation itself should be database independent as it does not rely on any MySQL/MyISAM specific features of SQL (but this has not been tested).

Implementation language The time/memory crucial parts of the system were implemented in C; the others which are not so demanding on resources, and also to comply with the research community standards, were implemented in Java or Python. Particularly the ESA-based Vector Search Engine is implemented in C along with its plug-in for MySQL. The script for building the index for ESA Vector Index Search Engine was implemented in Python because the index building is often performed from a text terminal and Python is more easily adjustable to provisional needs than Java. The rest of the system is implemented in Java.

Classifier After series of experiments with different types of classifiers using WEKA [10], and also for a good performance in a similar task [18], SVM has been selected as the classifier for the links.

Toolkits and Libraries

- Apache Lucene (http://lucene.apache.org/)
- Apache Commons (http://commons.apache.org/)

- Google Code Libraries for Java 1.5+ (http://code.google.com/p/guava-libraries/)
- Trove (http://trove.starlight-systems.com/)
- XStream (http://xstream.codehaus.org/)
- MySQL Connector (http://www.mysql.com/products/connector/)
- Freemaker (http://freemarker.sourceforge.net/)
- Weka (http://www.cs.waikato.ac.nz/ml/weka/)
- JWLP (http://code.google.com/p/jwpl/)
- LibSVM (http://www.csie.ntu.edu.tw/~cjlin/libsvm/)
- Weka Wrapper (http://code.google.com/p/weka-wrapper/)

4.1 Data Model

4.1.1 MediaWiki Database Layout

MediaWiki database layout consists of three tables:

- page (page_id, page_title) – list of pages with their titles
- revision (rev_id, rev_page, rev_text_id) – connects pages and texts (foreign keys: rev_page to page.page_id, rev_text_id to text.old_id)
- text (old_id, old_text) – list of texts
- langlinks (ll_from, ll_lang, ll_to) – list of cross-lingual links, that group together same articles in their different language versions

4.1.2 Vector Binary Format

The binary format which is used to represent the ESA vectors and ESA index search results has the following structure:

```
[number of dimensions]{[dimension id][concept value]}*
```

number of dimensions (4 byte big-endian unsigned integer) – information about the number of dimension-value pairs that follow

dimension id (4 byte big-endian unsigned integer) – identification number of the dimension

dimension value (4 byte IEEE 754 float) – value of the dimension

4.2 Cross-lingual Link Discovery System

The Cross-lingual Link Discovery System is composed of several modules that fit together as illustrated on **??**. Two classes of approaches to Link Discovery are structured into two subsections where their parts are discussed. The system has two modes, one for learning the classifier (decider) and the other one for actual link discovery.

4.2.1 Link Discovery: Methods Purely Based on Semantic Similarity

First approach towards Link Discovery is based on suggesting only a list of semantically similar documents, obtained in different ways. Namely CL-ESADirect, CL-ESA2Similar, CL-ESA2ESA. All of them are defined and described in Subsection 3.5.1. Their implementation largely uses the supportive tools later described in this chapter and is therefore very straightforward. Each of the methods first retrieves a list of semantically similar documents to the input document using the ESA Vector Search Engine (the search vector for the engine is built as described in Subsection 2.3.4). Then, each method makes different use of it:

CL-ESADirect The list of retrieved documents is used as the link suggestions.

CL-ESA2Similar The ESA vectors of first k documents are used. Then they are averaged and the documents that correspond to the dimensions with the greatest values are taken as the link suggestions.

CL-ESA2ESA The ESA vectors first k documents are used. Then they are averaged and the resulting vector is used for another search in the ESA Vector Search Engine for another semantically similar documents (this time the mapping of ESA Vector to the other language's concept space is not done). The resulting similar documents list is then used as the link suggestions.

4.2.2 Link Discovery: ESA LinkBase Method

The second approach is based on retrieving links from the semantically similar pages. These links are then used as linking suggestions, but because it is a computationally intensive process, the method is not as straightforward as the previous approach. It proceeds in several steps which are described bellow.

Document Retriever

The ESA Vector Search Engine is used to retrieve the semantically similar documents from the document collection. First, the input document is projected into the ESA concept space, using the ESA Document Analyser, and its Concept Vector is obtained. Then, the ESA Vector Search Engine is utilised, via the MySQL plug-in, to search for similar documents in the document index. As a result, the desired number of semantically similar documents is returned.

LinkMap Extractor

A set of semantically similar documents produced by the Document Retriever is used and existing links are extracted from them. The link extraction from raw Wikipedia articles is facilitated by the JWPL library [26]. The result of this module is a list of pairs `(<link text>, <target document id>)`.

LinkBase Builder

The output of the LinkMap Extractor module is taken and inserted into database.

4.2.3 Candidate Finder

The input document is searched and attempts are made to find the appropriate position for the suggested links. The link suggestions are passed here in form of a database table, built by one of the aforementioned Link Discovery methods. The pseudo-code of the Link Placing algorithm is in Algorithm 1.

4.2.4 Candidate Scorer

A set of features are assigned to each link. These features are described and defined in Subsection 3.5.3. The ESA Document Analyser is used to obtain Concept Vectors for similarity computation between the link text, source document and the target document. For other features the Generality Mapper and Link Frequency Index are used.

Generality Mapper

Generality Mapper assigns a Wikipedia article its generality. It is implemented as a database table of (article; generality) pairs. The generality value is based on the category tree of Wikipedia and originally used by [18] in their `wikipedia-miner` toolkit [17]. In our system it is implemented as a MySQL database table with two columns (page_id, generality).

Link Frequency Index

An index is used to determine how many times a particular text was used as a link in a document collection. In our system it is implemented as Lucene index with the search capability (due to large number of items it proved that it is better to use Lucene than a database table).

4.2.5 Candidate Picker

The last step of the link suggestion process is the candidate selection. A SVM decider and the link features are used to select the correct link candidates from the list of scored link candidates. Although, due to low recall of the link classifiers the list of selected links is short, therefore it needs to be complemented. We chose to do it by filling in links from the candidate suggestions list by sorting it in their decreasing semantic similarity order. Semantic similarity for the order is that of the input document with the link target document. This way a reasonable number of links for any document can be suggested giving satisfactory results.

Decider Trainer

The WEKA library and the Weka Wrapper library are utilised to build a SVM decider for classification of links. The output of the decider is a confidence coefficient in the interval $< 0; 1 >$ saying whether a link is valid or not.

4.2.6 NTCIR Submission Generator

The list of picked candidates from the Candidate Picker is taken and put into the format suitable for submission for the NTCIR2011:CrossLink competition.

4.3 ESA Vector Search Engine

ESA Vector Search Engine was built to facilitate fast search for similar documents in a huge document collection. The main aim was to conserve as much memory as possible while providing the fastest possible search capabilities.

The engine consists of three parts. First of them computes ESA vector for each of the documents in the document collection and stores them in a database. The second one analyses those ESA vectors and builds the vector index which is afterwards used by the third part for searching. The first two parts need to be used only once for creating the index. It is the third part that is then repeatedly used to retrieve the most semantically similar documents.

Standard POSIX function `mmap` was used to access files in a way that the operating system itself can decide how much it can load into memory. It also simplifies other access as the file presents itself transparently as a part of memory. The best performance is achieved when the whole index can be loaded into the memory (which is for English Wikipedia and ESA vectors of 100 dimensions around 4.5GB).

4.3.1 ESA Vector Computer

Each document of the given document collection is put into ESA Analyser which produces a Concept Vector which is stored in the database. Before putting the vector into database, trimming is done from computational and storage reasons. More about effects of trimming ESA vectors is discussed in Section 5.7. The whole process is illustrated in Algorithm 2. The function `esa_analyze(document)` which is called in the body of the algorithm is described in Algorithm 6 and computes ESA vector for the given document.

Algorithm 2 ESA Vector Computer

```
1: procedure esa_vector_computer(docs):
2: for d ∈ docs do
3:     esa_vector ← esa_analyze(doc);
4:     store(db, esa_vector);
5: end for
```

4.3.2 Index Builder

Index Builder prepares an inverted index of ESA vector dimensions. It iterates through each document of the given document collection, reads its ESA vector from database and puts it into index. The index indexes vectors according to the non-zero dimensions that they contain so that the look-up of documents by dimension id is fast. Algorithm 3 describes in pseudo-code how the index is constructed.

Algorithm 3 uses auxiliary function `write(file, data)` that writes the given data to a file. Also, the function `esa_analyze(document)` computing ESA vector of the input document is used (described in Algorithm 6).

Index Files

The following files are built to provide the document search capabilities based on the non-null dimension of ESA vectors of the documents:

Document Vector Storage This file contains all ESA vectors in the binary format.

Document Vector Index This file provides a pointer to the Vector Storage file for each document.

Dimension Storage This file contains for each dimension a list of documents where it figures.

Dimension Index This file provides a pointer to the Dimension Storage file for each dimension.

Dimension Mapping This file provides a mapping between dimension id (which is often not a continuous block of numbers) and its internal representation (which is from a continuous block of numbers).

Algorithm 3 Index Builder Algorithm

```
1: procedure build_index(docs):
2: var map{}; # map of dimensions to a list of documents which contain them
3: var f_document_vector_storage; # file for Document Vector Storage
4: var f_document_vector_index; # file for Document Vector Index
5: var f_dimension_storage; # file for Dimension Storage
6: var f_dimension_mapping; # file for Dimension Mapping
7: var f_dimension_index; # file for Dimension Index
8: for doc ∈ docs do
9:    esa_vector ← esa_analyze(doc);
10:   for dimension ∈ esa_vector do
11:      map[dimension] ← map[dimension] ∪ {doc};
12:      write(f_document_vector_storage, esa_vector);
13:      write(f_document_vector_index, doc);
14:   end for
15: end for
16: sort(map); # sort map according to the dimensions
17: for dimension ∈ map do
18:   write(f_dimension_mapping, dimension);
19:   write(f_dimension_index, tell(f_dimension_storage));
20:   for doc ∈ map[dimension] do
21:      write(f_dimension_storage, doc);
22:   end for
23: end for
```

Time and Space Complexity Analysis

The Algorithm 3 is in the time complexity class $O(N)$. This is with respect to the number of documents being indexed if a hash-table is used as the implementation for `map`. Number of dimensions of the ESA vectors is fixed (e.g. 100 in our case), and the look-up operation `get_esa_vector` is constant as in the implementation the documents are retrieved along with their ESA vectors.

Space complexity class of the Algorithm 3 is $O(N)$, as the space can increase only linearly with the length of the ESA vectors trim (e.g. 100 in our case).

4.3.3 Index Searcher

As an input, a vector, index path and a number of documents to retrieve are given. Using the vector index a subset of documents to consider is selected. Then the Cosine similarity is computed between each of the documents in this subset and the input vector. Cosine similarity computation yields a real number (similarity measure) for each of the considered documents. The documents are then sorted according to this number and the required number of the best scoring documents is returned as a result.

Index Searcher operates according to Algorithm 4. The function `cosine_similarity(v1, v2)` proceeds as described in Algorithm 5. Function `scan(index file, item id)` in the algorithm performs binary search for a specific item over a file with the index. Function `read(file, position)` reads data from a given position in a file. Function `get_docs(dimension)` returns a list of documents whose value in ESA vector for that dimension is greater than zero. Function `write(file, data)` writes the given data to a file. Functions `get_bit(var, bit_id)` and `set_bit(var, bit_id)` are used to read and set the particular bits of a given variable, respectively.

Algorithm 4 Index Searcher Algorithm

```
 1: procedure search_index(vector
                f_document_vector_storage, f_document_vector_index,
                f_dimension_storage, f_dimension_mapping,
                f_dimension_index):
 2: var history; # bit array saying which articles have been compared
 3: for dimension ∈ vector do
 4:   # get the internal dimension id;
 5:   dimension_id ← scan(f_dimension_mapping, dimension);
 6:   docs ← get_docs(dimension_id);
 7:   for doc ∈ docs do
 8:     doc_vector_id ← scan(f_document_vector_index, doc);
 9:     if get_bit(history, doc_vector_id) == 0 then
10:       doc_vector ← read(f_document_vector_storage, doc_vector_id);
11:       score ← cosine_similarity(vector, doc_vector);
12:       result[doc] ← score;
13:       set_bit(history, doc_vector_id);
14:     end if
15:   end for
16: end for
17: sort(result);
18: return result;
```

Time and Space Complexity Analysis

The time complexity class of Algorithm 4 is $O(N \cdot k)$. Binary search function `scan` belongs to $O(log(N))$. The document loop for each dimension can at the worst case turn out to proceed through all N documents, which means that in the worst case all documents have to be considered. The `cosine_similarity` function is in the complexity class $O(k)$ with respect to the number of non-null (untrimmed) dimensions.

Algorithm 5 Cosine Similarity Computation Algorithm

```
1: procedure cosine_similarity(v1, v2):
2: var values{};
3: var v1_norm ← 0;
4: var v2_norm ← 0;
5: for dimension ∈ v1 do
6:    v1_norm ← v1_norm + v1[dimension];
7:    values[dimension] = v1[dimension];
8: end for
9: for dimension ∈ v2 do
10:   v2_norm ← v2_norm + v2[dimension];
11:   result ← result + values[dimension] * v2[dimension];
12: end for
13: return  result / (sqrt(v1_norm) * sqrt(v2_norm));
```

The space complexity class is $O(N)$ with respect to the number of documents in the collection.

4.4 MySQL Plug-ins

MySQL plug-ins were built to facilitate faster document retrieval of semantically similar documents from database. This is ensured by limiting the communication between the database and the system. The first function computes the similarity between ESA vectors, therefore allows for database SELECTs that consider semantic similarity of the documents at the database layer. The second function is an interface to the implementation of the ESA Vector Search Engine which does not use any database data but only makes the communication with the engine easier [1].

Format of the vectors expected by the following methods is described in Subsection 4.1.2.

4.4.1 esa_simil(v1, v2)

The plug-in computes Cosine similarity between two ESA vectors that are passed to it as BLOB parameters using the Algorithm 5. The result is a real number.

4.4.2 esa_search(v, index_path, number_of_documents)

The plug-in searches the given vector index for the most similar documents to the document represented by the vector v. The algorithm used for searching is identical to Algorithm 4 as the plug-in itself is just a MySQL wrapper around the search function. The result is a binary string in the format used for ESA vector representation (Subsection 4.1.2).

[1] Effectively it spares us the trouble of having to devise a communication protocol and implementing an Internet server.

4.5 Loading Wikipedia Dump to Database

Throughout the development process it proved to be quite challenging to import the Wikipedia data from the supplied dumps [2] to the database. A number of tools exist to support this but only one tool proved itself to be satisfactory. The others often crashed due to an encoding error or were not compatible with the current Wikipedia dumps. We also encountered serious speed issues when trying to import data to the database which was running on a Windows machine.

The quickest and most convenient way how to import dumps into the database is:

1. Process the dump by the WikiXRay [3] which produces three `.sql` files for import into the database.
2. Create the target database.
3. Alter the MediaWiki database schema so that it does not contain any indexes.
4. Import the `.sql` files from the previous step by the standard `mysql` CLI.

4.6 Wikipedia Prepare

The Wikipedia Prepare is used to prepare the Wikipedia database for further usage. The MediaWiki database after the MySQL import finishes processing the WikiXRay `.sql` files is taken as an input. Then, database indexes are created, talk and other auxiliary pages deleted, the disambiguation and redirect pages identified and the concept mapping to a given version of Wikipedia is created. As the last step, a database table `page_concepts` is created. It contains only non-redirect, non-disambiguation, non-talk Wikipedia pages which can be used as the ESA background.

For building the concept mapping it is required that the MediaWiki table `langlinks` exists and is filled with appropriate entries. For identifying the disambiguation pages a pattern which denotes a disambiguation pages is needed (in most cases the pattern is `{{disambig` and `{{hndis`, but it can differ for some language versions of Wikipedia).

4.7 ESA Inverted Index Builder

ESA Inverted Index Builder prepares a collection of documents so that it can be used as the ESA Background for projecting documents into the Concept Space. The Concept Space is defined by this background document collection. The result of this building process is called the ESA Inverted Index. Inverted because each term is paired with information about a list of documents where the term occurs (as opposed to pairing each document with a list of its terms). Base of the code was adopted from the original proof-of-concept implementation of [5].

1. First, all documents of the background document collection are normalised (stemmed, stop-words are removed) and stored in a Lucene index.

[2] http://dumps.wikimedia.org/
[3] http://meta.wikimedia.org/wiki/WikiXRay

2. Then, the index is read and a file containing entries (`<term>; <document id>; <term tfidf>`) is created. Only terms occurring in more than a certain number of documents are included. This threshold is adjustable and influences the size of the created ESA Inverted Index. Consequently it even impacts the speed of document projection into the ESA concept space as the ESA Inverted Index is the foundation for the projection process.

3. Afterwards, the file is sorted according to the `<term>` field. Basically, this file contains almost [4] all terms from all documents from the background collection. Because the file is sorted according to the `<term>` field it is easy to obtain a list of documents where a term figures.

4. The list of documents for each term is filtered by a windowing function (introduced in [5]) that limits the number of documents that are assigned to each term.

After the windowing is done, building of the ESA Inverted Index is finished. Eventually it comprises two database tables `terms` and `index`. The former has records about IDF of all terms, while the latter holds a list of documents where the term occurred along with its TF-IDF there. The list of documents is recorded as a binary string, format of which is described in Subsection 4.1.2.

4.8 ESA Document Analyser

This part of the system takes care of projecting an input document into the ESA concept space. As a background the ESA Inverted Index built by Section 4.7 is used. ESA Document Analyser produces a Concept Vector for the input document as an output.

Steps

1. The input document is tokenized, the tokens are stemmed, stop-words removed.

2. Then, the ESA Inverted Index is engaged for building the Concept Vector. Each token is taken and used as a look-up key for the ESA Inverted Index. The token's IDF is obtained from the index and used to produce its TF-IDF value.

3. Afterwards, the ESA Inverted Index is used to obtain a list of documents (concepts) where each token appeared with its TF-IDF (TF-IDF of the token here is the one that was computed in the background collection). The obtained TF-IDF value is then multiplied with the TF-IDF of the token in the current document and added to the score for the particular concept.

4. At the end, the Concept Vector is created from the values that were recorded for each of the concepts.

The algorithm is more formally written in Algorithm 6. Function `tokenize(document)` splits the document into tokens and stems them. Function `get_esa_bg_vector(token)` looks up the token in the ESA background inverted index and retrieves the vector of background documents where the token appeared along with corresponding TF-IDF values

[4] This depends on the term frequency threshold that was applied before the terms were written to the file.

of the given term in that document. Function `tfidf(token, document)` computes the TF-IDF value of the token in the document given the ESA background collection.

Algorithm 6 ESA Document Analyser

```
1: procedure esa_analyze(doc):
2: var values{};
3: tokens ← tokenize(doc);
4: for token ∈ tokens do
5:     bg_vector ← get_esa_bg_vector(token);
6:     for (concept, value) ∈ bg_vector do
7:         values[concept] ← values[concept] + value * tfidf(token, doc);
8:     end for
9: end for
10: sort(values);
11: return values;
```

Chapter 5

Evaluation and Experiments

As a whole, our systems has been evaluated by the standard precision-recall metrics (Subsection 2.4.1) by the methodology described in Subsection 2.4.5 for evaluating performance of the system. Other than that, we analysed how well is Explicit Semantic Analysis able to identify the counterpart of a given page across languages. In addition, we measured the agreement between human annotators and computer annotation, and between different human annotation. And also, we measured the influence of cutting the ESA vector of a document on the precision of similarity computation.

5.1 Data

Wikipedia has been used as a corpus for testing our methods, because:

- Wide variety of articles is available in many language versions (e.g. articles on one specific topic are available in many languages, with hopefully very similar content).
- The articles are well-interlinked and the interlinking result have been approved by a large community of users.
- Different language versions of an article are explicitly mapped together (therefore we can exploit the mapping).

We have conducted experiments with the English, Spanish, Chinese and Czech language versions of Wikipedia. The language selection has been motivated by an intention to test the methods on different article-base sized collections and also to take part in the NTCIR2011:CrossLink competition.

The English Wikipedia contains 3,665,185 articles. The Spanish version of Wikipedia contains 764,095 articles, the Czech version is much smaller and contains only 196,494 articles. And the Chinese Wikipedia contains 318,736 articles

In all experiments the testing topics were excluded from the background corpus of the Explicit Semantic Analysis.

5.2 Auxiliary Definitions

5.2.1 Ground Truth

We define the ground truth of a document collection as a set of already existing links in it, denoted Π_D (discussed in Subsection 3.4.1).

$$\mathrm{GT} = \Pi_D \tag{5.1}$$

A multi-lingual ground truth is defined as the union of a subset of the ground truth of one document collection, and the ground truth of the other one. The subset is defined as a set of links in which both document, the source document and the target document, are mappable to the other document collection:

$$GT_{D_2|D_1} = \{(d_{1*}, d_{2*}) | (d_1, d_2) \in \Pi_{D_1}, d_{1*} = \rho_{D_1 \to D_2}(d_1), d_{2*} = \rho_{D_1 \to D_2}(d_2)\} \tag{5.2}$$

The joint ground truth for the document collections D_1 and D_2 is following:

$$\mathrm{GT}_{\mathrm{D_1,D_2}} = GT^*_{D_2} \cup \Pi_{D_2} \tag{5.3}$$

5.2.2 Link Hit

A link from a document d_a to a document d_b is is evaluated as a hit if and only if it belongs to the ground truth:

$$\mathrm{hit}(d_a, d_b) \Leftrightarrow (d_a, d_b) \in GT \tag{5.4}$$

If a link is evaluated against a ground truth of a different language version of the collection, the cross-lingual mapping function must be engaged. Consequently, a link from a document d_a to a document d_b ($d_a, d_b \in D_1$) is considered as a hit in the ground truth GT_{D_2} of the document collection D_2, if and only if the documents that we acquire by the cross-lingual document mapping function are linked in the collection D_2:

$$\mathrm{hit}(d_a, d_b) \Leftrightarrow (\rho_{D_1 \to D_2}(d_a), \rho_{D_1 \to D_2}(d_b)) \in GT_{D_2} \tag{5.5}$$

5.3 Cross-lingual Link Discovery Evaluation #1

This evaluation was conducted on four different configurations of the cross-lingual link discovery system. The configurations differ in the Link Discovery component, which is either ESA link base, terminology, or their combination.

5.3.1 Evaluation Setup

Evaluation methodology of the cross-lingual linking system was taken from the NTCIR:CrossLink Task evaluation [20]. Precision and recall of the discovered links were measured against the ground truth extracted from Wikipedia.

We have evaluated the following methods, all described in Section 3.5:

1. ESA link base + Terminology + SVM
2. Terminology + SVM

Method	MAP
ESA LinkBase + Terminology	0.26
ESA LinkBase	0.251
Terminology	0.127
ESA LinkBase + Terminology (without CL map)	0.041

Table 5.1: Results of the Cross-lingual Link Discovery. Link discovery from English document to a collection of Chinese documents.

3. ESA Link Base + SVM
4. ESA Link Base + Terminology + SVM, without the cross-lingual map between the document collections [1]

Data

The data on which our cross-lingual link discovery system was tested is a set of 25 English Wikipedia articles picked by the organisers of the NTCIR2011:CrossLink competition. The target document collection for links is the Chinese version of Wikipedia. All links and supporting information are cleared from the testing English articles. The remaining link structure is kept.

5.3.2 Results

At the time of this report writing, the official results were unavailable so a comparison with the systems of other contestants is missing. In Figure 5.1 the P-R graph of the performance of the methods is presented, and in Table 5.2 the MAP metric of the evaluated methods is summarised. This was generated by the official tools and the official ground truth.

In the results the method that does not use the cross-lingual mapping between the document collections but has to compute it itself performed the worst. On the other side, the best results were achieved by utilising both, the ESA link base and the terminology dictionary. Without the terminology dictionary, this method performed only a little bit worse. So we can conclude that the link base built from the semantically most similar pages pretty much covers all relevant links.

Because our methods are currently unable to find links to pages that do not have the cross-lingual mapping back to the source collection, we counted the number of such links. Our measurements show that such links make up in average only 12 % of the ground. Therefore, if a precise way to discover those pages was found our methods could have been improved.

5.4 Cross-lingual Link Discovery Evaluation #2

The second evaluation was conducted to see how the systems perform when the links point from a document in a document collection that is poorer in content to documents in a

[1] This means that the system had to attempt first to find the cross-lingual counterpart for a document before it suggested it as a link. As opposed to the other three methods, which know exactly the cross-lingual mapping.

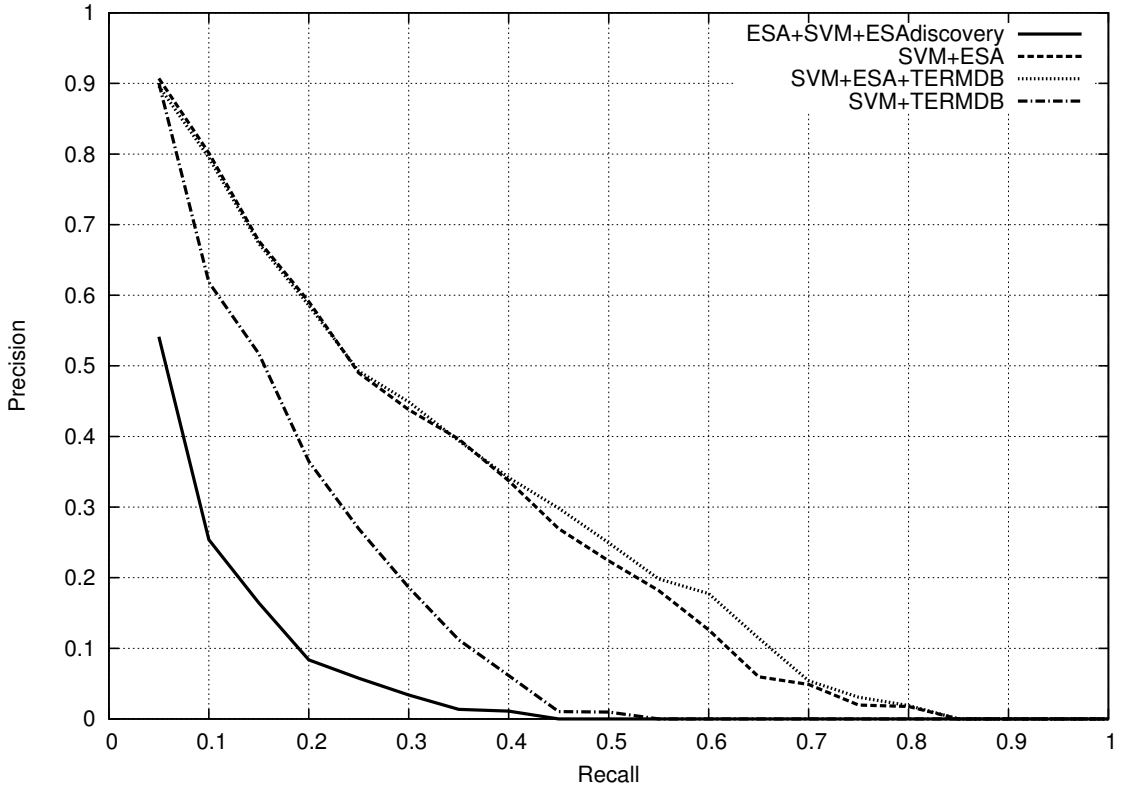

Figure 5.1: P-R graph comparing the 4 cross-lingual link discovery methods applied to a set of 25 testing English Wikipedia articles linking to Chinese Wikipedia.

document collection that is richer in content. This is also an important direction for cross-lingual link discovery as the target language version is more likely to contain relevant information not available in the source language, or provide the information in higher quality.

5.4.1 Evaluation Setup

We chose to test the system to generate links in Spanish to English, and Czech to English directions. As a ground truth we took all links that are cross-lingually mappable between the two given languages. The four methods that we evaluated are described in Subsection 3.5.1, namely CL-ESADirect, CL-ESA2Links, CL-ESA2Similar and CL-ESA2ESA. In this experiment the Link Placement and the Link Classification parts of the system were not engaged. Only the performance of the Link Discovery part was measured.

Data

The data for this experiment are equivalent to the data described in Subsection 5.5.1.

5.4.2 Results

The results of the four different methods, for the two cases are presented in Figure 5.2. As expected, it shows that suggesting links purely based on semantic similarity of documents

Spanish	
Method	MAP
CL-ESA2Links	0.091
CL-ESA2Similar	0.023
CL-ESA2ESA	0.014
CL-ESADirect	0.016

Czech	
Method	MAP
CL-ESA2Links	0.085
CL-ESA2Similar	0.027
CL-ESA2ESA	0.011
CL-ESADirect	0.016

Table 5.2: Results of the Cross-lingual Link Discovery. Link discovery from a Spanish/Czech document to a collection of English documents.

is inferior to the method exploiting the existing link structure.

5.5 Agreement Measurement

To assess the subjectivity of the link generation task and to investigate the reliability of the acquired ground truth, we have compared the link structures of different language versions of Wikipedia.

5.5.1 Evaluation Setup

The experiment was carried out on two language pairs: Spanish to English and Czech to English. We will denote the source language L_1 and the target language L_2. Also, other terms and identifiers are those introduced in Section 3.5. The input document sets are:

- Let D_1 be a document collection written in L_1. Let $D_{1*} \subseteq D_1$ be its subset, $|D_{1*}| = 100$. It was selected in a semi-random way from those pages that are cross-lingually mappable between different language versions of Wikipedia (by the partial function $\rho_{D_1 \to D_2}$, discussed in Subsection 3.4.2).
- Let D_2 be a document collection written in L_2 from which the link targets are selected. In our case, this collection contains all (3.8 million) Wikipedia pages in English.

Kohen's Kappa

A common way to assess inter-annotator agreement between two raters in Information Retrieval is using the Cohen's Kappa. This is typically calculated as:

$$\kappa = \frac{Pr(a) - Pr(e)}{1 - Pr(e)},$$

- $Pr(a)$ is the relative observed frequency of agreement, and
- $Pr(e)$ is the hypothetical probability of chance agreement

In our experiment, $Pr(a)$ is computed as the ratio of agreement and the total number of links (C_i denotes the count of items in the class i):

$$Pr(a) = \frac{|C_{YY}| + |C_{NN}|}{|C_{YY}| + |C_{YN}| + |C_{NY}| + |C_{NN}|} \tag{5.6}$$

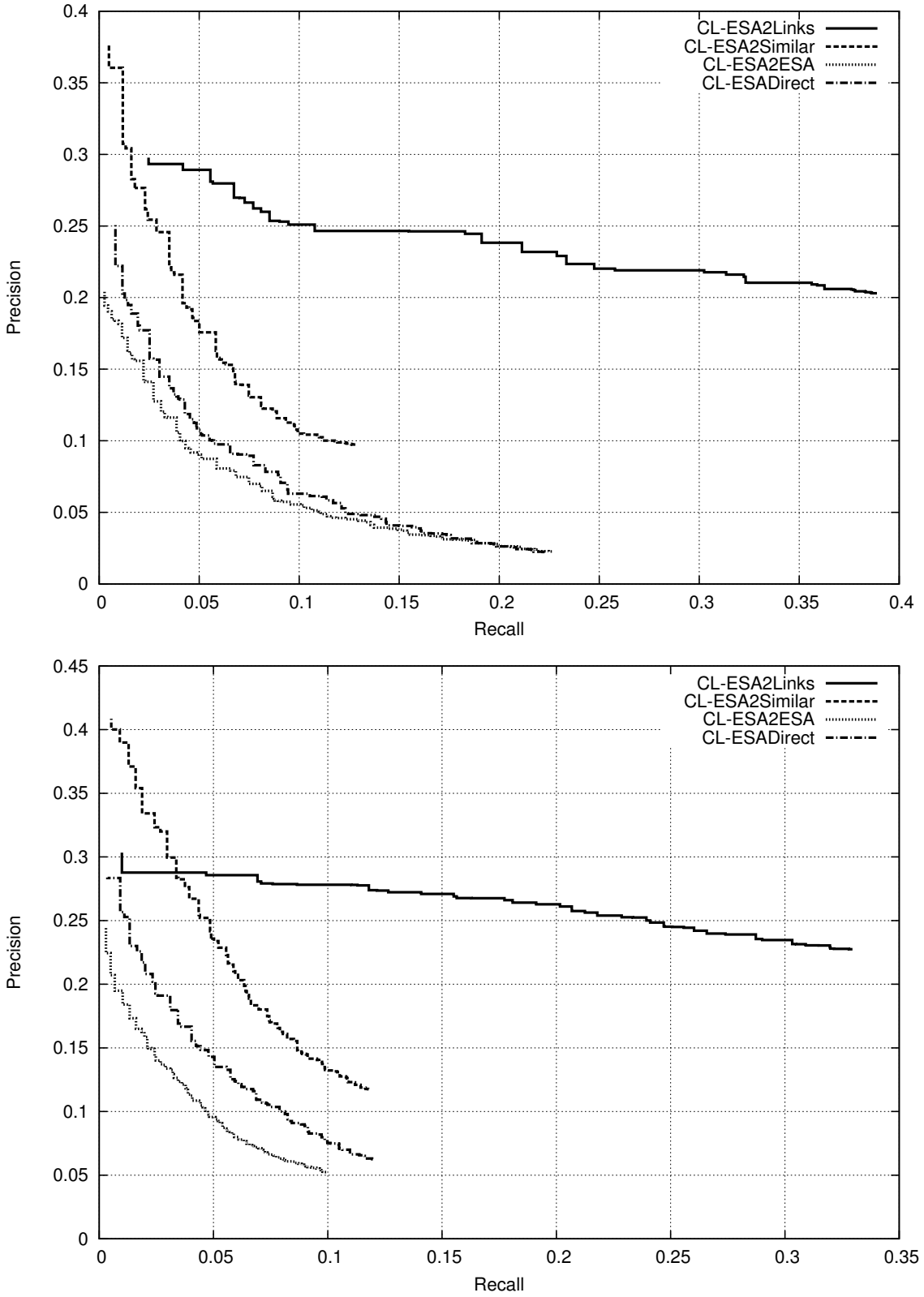

Figure 5.2: The precision ($y - axis$)/recall (x-axis) graphs for Spanish to English (up) and Czech to English (bottom) cross-lingual link discovery methods.

Spanish vs English			
	Y_{en}	N_{en}	N/A_{en}
Y_{es}	5,563	10,201	3,934
N_{es}	15,715	539,299,641	99,191,766
N/A_{es}	5781	321,326,145	0
Czech vs English			
	Y_{en}	N_{en}	N/A_{en}
Y_{cz}	4,308	8,738	2,194
N_{cz}	12,961	392,411,445	7,501,806
N/A_{cz}	9,790	356,532,740	0

Table 5.3: The agreement of Spanish and English Wikipedia and Czech and English Wikipedia on their link structures for all 100 pages in D_{1*}. Y - indicates yes, N - no, N/A - not available/no decision

Although, if the agreement is measured on all four cases (YY, YN, NY, NN), the agreement is very close to 1. This is due to the large agreement on the negative examples. But since the agreement on the negative examples is not important for our study, we will neglect the NN class and estimate $Pr(a)$ as follows:

$$Pr(a) = \frac{|C_{YY}|}{|C_{YY}| + |C_{YN}| + |C_{NY}|} \tag{5.7}$$

5.5.2 Results

We have iterated over the set of documents from D_{1*} and recorded for each document d_c from D_1 and D_2 [2] whether:

1. the document d_c is a hit for the document collection D; therefore, the class Y_D was assigned.
2. the document d_c is not a hit for the document collection D; therefore, the class N_D was assigned.
3. the document d_c does not exist [3] in the document collection D; therefore, the class N/A_D was assigned.

As this was done for both document collections, each link has two classes. Counts of links in each of the classes after applying this method on the document collections D_{en}, D_{cs} and D_{es} is printed in Table 5.3.

The probability of a random agreement or a random appearance of an item in one of the three mentioned classes is extremely low, because the probability of a link connecting any two pages is roughly[4]:

$$p_{link} = \frac{\#\ of\ links}{(\#\ pages)^2} = \frac{78.3M}{3.2M^2} = 0.000007648 \tag{5.8}$$

[2]The document mapping function $\rho_{D_1 \to D_2}$ was used to identify the equivalent documents in both collections.

[3]As shown in Figure 5.3, a subset of Wikipedia pages cannot be mapped to other language versions because the page does not exist in the other language.

[4]Following the official Wikipedia statistics.

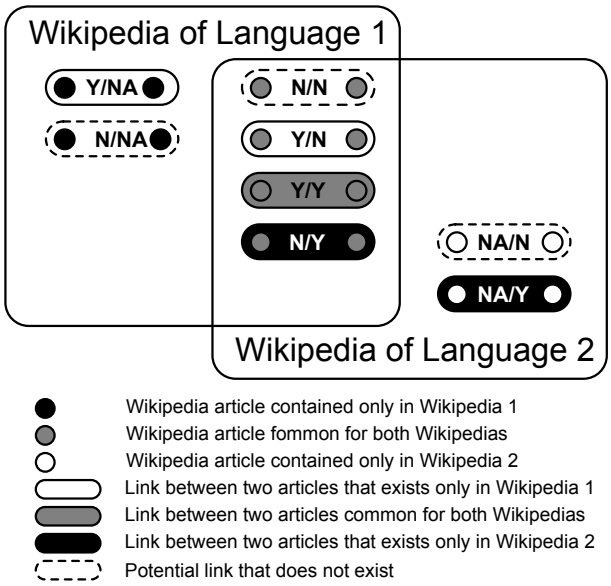

Figure 5.3: Visualisation of link agreement/disagreement/not available for two language versions of Wikipedia collections. The text inside the link oval denotes the decision for the first Wikipedia and for the second Wikipedia, respectively.

Thus, the hypothetical number of items appearing in the C_{YY} class by chance is:

$$N_{YY\ chance} = p_{link}^2.(|C_{YY}| + |C_{YN}| + |C_{NY}|) \sim 0 \tag{5.9}$$

In the classes C_{YN} and C_{NY} it is:

$$N_{YN/NY\ chance} = (1 - p_{link}).(|C_{YY}| + |C_{YN}| + |C_{NY}|) \sim 0 \tag{5.10}$$

In all cases, this value is well below 1. This means that it is unlikely that a single item in Table 5.3 in the three classes used in our calculation appears by chance. Hence $P(e)$ is close to 0.

Kohen's kappa correlation coefficients for:

- the agreement between English and Spanish:

$$\kappa_{en,es} = \frac{5,563}{31,479} = 0.177$$

- the agreement between English and Czech:

$$\kappa_{en,cz} = \frac{4,308}{26,007} = 0.166$$

This indicates a relatively low inter-annotator agreement. The fact that that such a low agreement has been measured is interesting, particularly because the link structure in Wikipedia is a result of a collaborative effort of multiple contributors so general cohesion is expected. In addition, it supports our arguments about the unsuitability of Wikipedia being the ground truth for cross-lingual linking evaluation, discussed in Section 5.8.

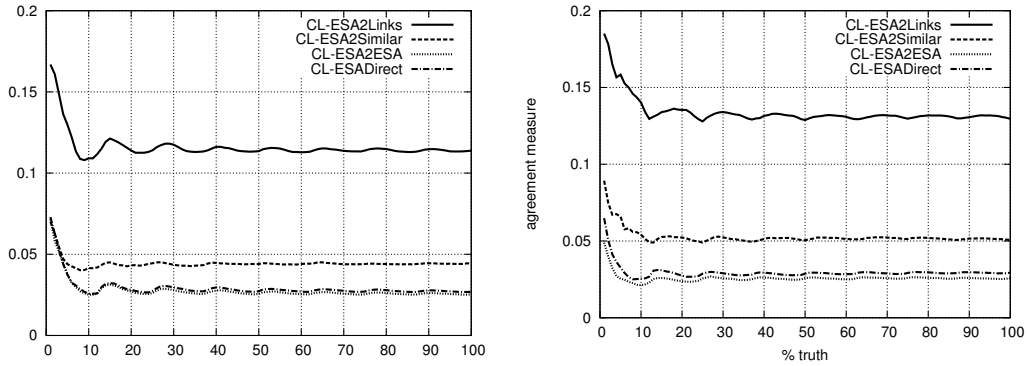

Figure 5.4: The agreements of the Spanish to English (left) and Czech to English (right) CLLD methods with $GT_{es,en}$ and $GT_{cz,en}$ respectively. The y-axis shows the agreement strength and the x-axis the number of generated examples as a fraction of the number of examples in ground truth.

Motivated by the previous findings, we have calculated the agreement between the output of our methods and the link graphs present in different language versions of Wikipedia. We were especially interested whether the agreement is significantly different from the agreement measured between different language versions of Wikipedia above.

For every testing document from D_{1*} we took the output of our methods, sorted according to confidence, and trimmed it at the number of links which are in the testing document's ground truth (i.e. if a particular document is linked in Wikipedia to 57 documents, we took the first 57 links discovered by our methods). Then, we have measured the agreement for each document and averaged the agreement values for the whole collection.

The results of the experiment for Spanish to English and Czech to English cross-lingual link discovery are shown in Figure 5.4. The figures suggest that CL-ESA2Links achieved a level of agreement comparable to human annotators. A reasonable level of agreement has also been measured for CL-ESA2Similar, especially for the first 10% of the generated links. CL-ESADirect and CL-ESA2ESA exhibit a lower level of agreement.

5.6 Cross-lingual Counterpart Identification

To explore the properties and behaviour of the Explicit Semantic Analysis we analysed how well it can identify the cross-lingual counterpart of a given document in a document collection. The evaluation was conducted on Wikipedia.

5.6.1 Evaluation Setup

Our document retrieval engine ranks the documents in a document collection according to their semantic similarity to the query. For an input document d_1 from a document collection D_1 , and it's language counterpart d_2 from a document collection D_2, we measure the rank of d_2 in the results given d_1 as an input.

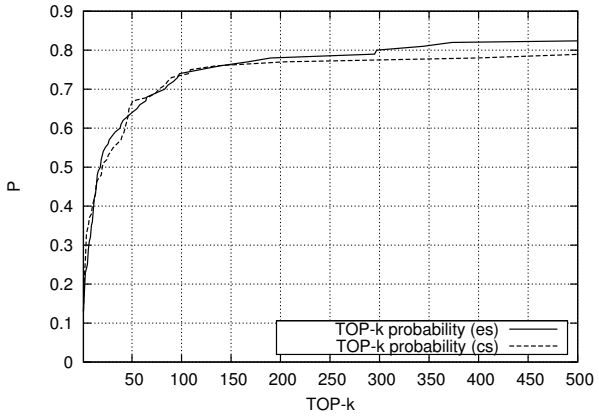

Figure 5.5: The graph shows the the relative frequency (y-axis, defined in Equation 5.12) of finding the language counterpart of a given document in the top k retrieved documents (x-axis), using the cross-lingual ESA retrieval engine.

Data

The test collection of documents comprises of 100 Wikipedia pages and is the same as the one in the previous experiment Subsection 5.5.1.

Measure

For each document d from the document collection D_{1*} the rank $rank(d)$ is recorded. It is the number of documents that were ranked better by the ESA retrieval engine than the document d_2:

$$rank(d) = |\{d : d \in D_2, score(d) > score(d_2)\}| \tag{5.11}$$

For presenting the results the following TOP-k measure is used:

$$TOP(k) = \frac{|\{d : d \in D_{1*}, rank(d) \leq k\}|}{|D_{1*}|} \tag{5.12}$$

It is a relative frequency of documents in the document collection D_{1*} that will have its correct language counterpart in the best k retrieved documents.

5.6.2 Results

The results are presented in Figure 5.5. They show that in 13 % cases, the language counterpart is the first retrieved document, and in 40 % cases the language counterpart is present in the first 10 retrieved documents. In the future, it would be very interesting to see what results can be achieved on document collections other than Wikipedia. We believe that if the documents were closer to being its translations [5] the performance of the counterpart identification would improve.

[5] In Wikipedia, the documents often significantly differ in content, due to the quality of the document, and cultural differences and background of the writers.

5.7 Explicit Semantic Analysis Vector-length Influence on Document Retrieval

Dimensionality of Explicit Semantic Analysis vectors is usually high (4 million dimensions) which is for further operations over them computationally expensive. That is true for both, computing similarity between individual vectors as well as for using the dimensions for selecting a subset of potentially relevant documents as discussed in Section 3.6. Here we have tested the influence of neglecting certain dimensions of ESA vector on the results of similarity computation. Correlation between the results of computation with trimmed and with original vectors was measured.

5.7.1 Evaluation Setup

Correlation of similarity results is measured by comparing lists of documents ranked according to similarity with a certain document. Three different document data-sets were used:

- relevant documents (to the testing document)
- randomly selected documents
- 50 % relevant documents, and 50 % randomly selected documents

Spearmen's Rank Correlation Coefficient

As a measure for comparing the effect of ESA vector cut on the rank of retrieved documents the Spearmen's rank correlation was used. It is a special case of the Pearson's r coefficient for ranked data which compares differences between ranks of observed items [19]:

$$r_s = 1 - \frac{6 \sum_i D_i^2}{N(N^2 - 1)} \tag{5.13}$$

- N – number of documents
- D_i – difference between the rank of i-th document

By our evaluation methods two lists of ranked documents are obtained: (i) the list of ranked documents which were ranked using the full ESA vector, (ii) the list of ranked documents which were ranked using a trimmed ESA vector. Correlation between those two lists is then computed.

Data

As a query the Wikipedia article about Linux is used. As the set of semantically similar documents, 200 documents from English Wikipedia which contain the word "Linux" in their title were used. The randomly picked documents were a set of 200 documents selected out of the English Wikipedia. The mixed data-set consists of 100 documents from the first data set and 100 documents from the second data set.

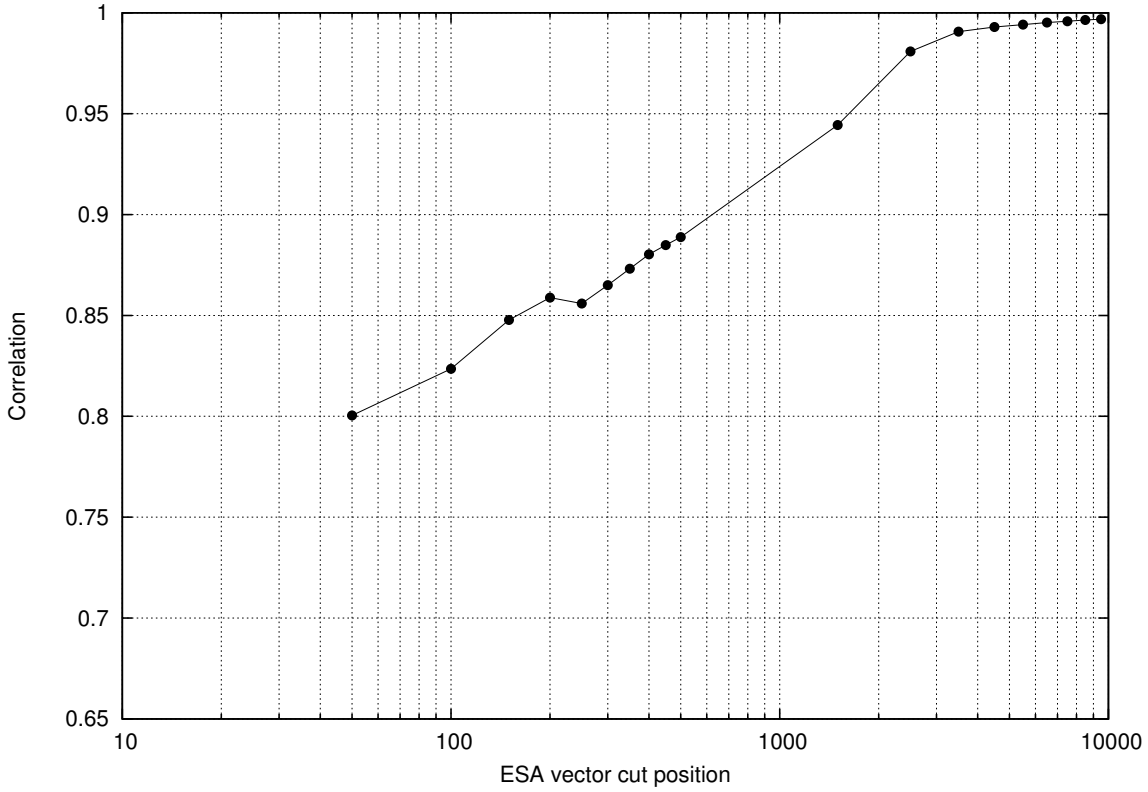

Figure 5.6: Relevant documents. The graph shows the relationship between correlation of the ranked document list (y-axis) and the ESA vector length (x-axis).

5.7.2 Results

The results for all three data sets are presented in Figure 5.6, Figure 5.7 and Figure 5.8, respectively. The correlation of the results in all three data sets is high (lowest with the random documents data set, highest with the half-half documents data set), which shows that the document ranking using the ESA similarity measure is mostly prevailed even if some dimensions are forgotten. As a result, we conclude that the trade-off of cutting the ESA vector is reasonable and should not have any major impact on the performance of the methods that use the ESA vector similarity for ranking documents. Also, our cursory experiments with the length of ESA vector in the cross-lingual link discovery system state the same.

5.8 Cross-lingual Evaluation Errors

The evaluation methods for the cross-lingual link discovery system were used at NTCIR2011:CrossLink for the first time and offer themselves to improvement. Mainly the automatic evaluation method introduces not negligible errors to the evaluation. This stems from the fact that the evaluation was basically taken from the evaluation methodology for the mono-lingual link discovery systems and just a little tweaked to work for the cross-

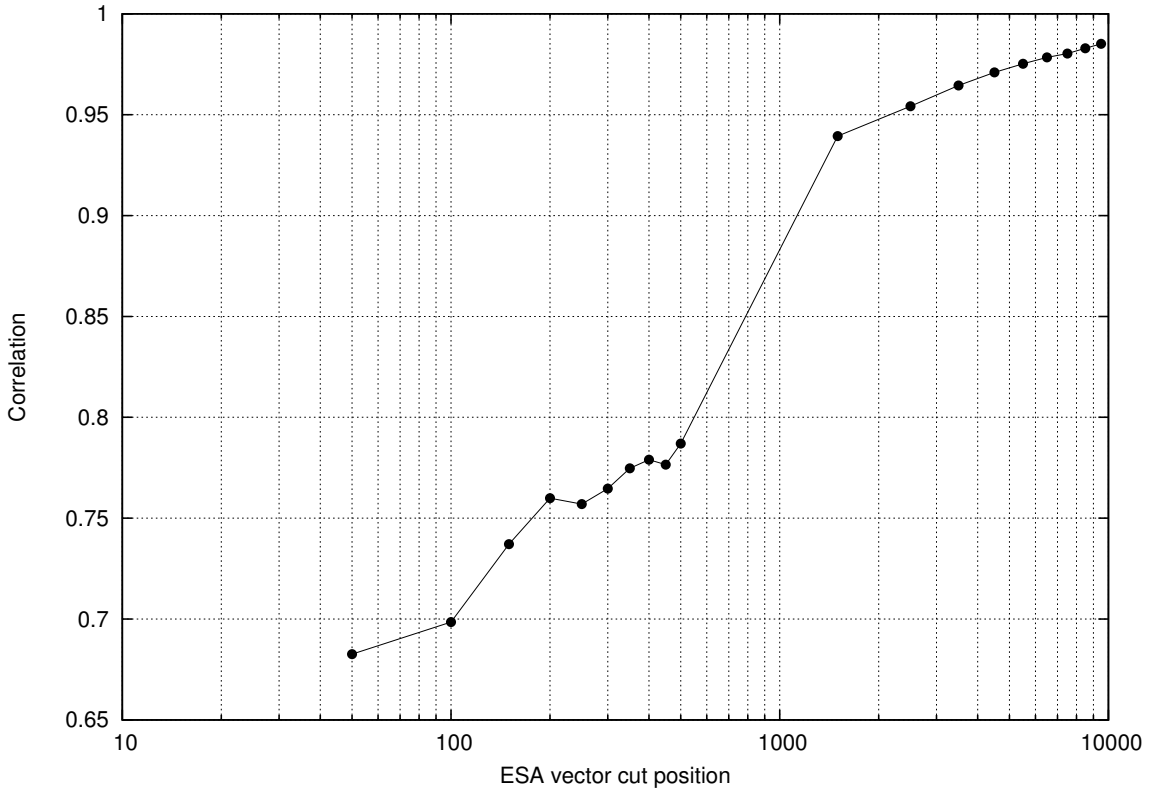

Figure 5.7: Randomly selected documents. Results for The graph shows the relationship between correlation of the ranked document list (y-axis) and the ESA vector length (x-axis).

lingual link discovery systems. It is not optimal, but as far as we know, no other standard evaluation procedure for such systems has been established yet.

1. First kind of error stems from inequality of the content of articles in different language versions of the Wikipedia. An article on the same topic in one language can contain entirely different content than an article in another one (e.g. article about the "public transport" contains specific information about public transport in a particular country – in Czech Wikipedia there is the description of underground in Prague, whereas in the French one there are details about various TGV trains specific for France). Therefore, if the existing link structure of Wikipedia is exploited for building the ground truth, an objectively relevant link is often evaluated as non-relevant.

   This is illustrated by our agreement measurement experiments between human annotators on the Wikipedia.

2. Link incompleteness of the Wikipedia gives rise to the second kind of error. The cross-lingual link discovery system might offer useful and relevant links that were just omitted by the human annotators (= people who create the Wikipedia articles and inter-link them).

3. The third possible source of evaluation error comes from the incompleteness of the cross-lingual links on the Wikipedia. By our experiments we measured that only

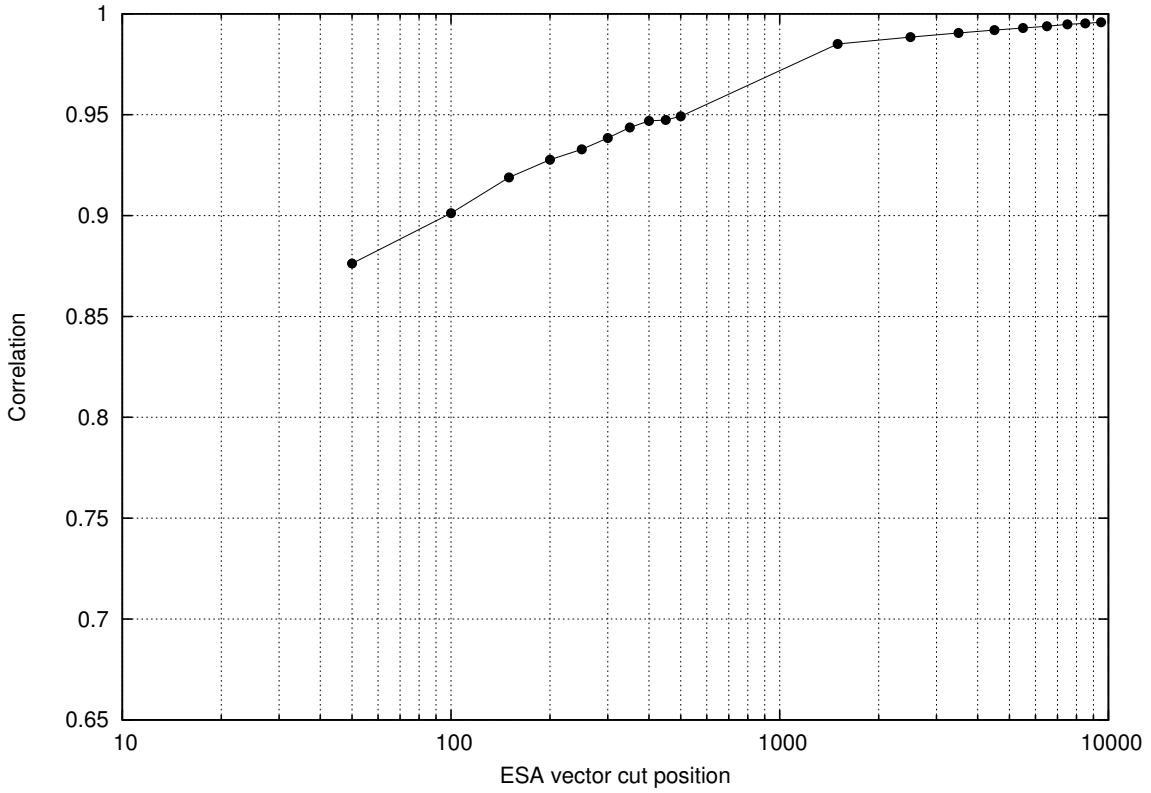

Figure 5.8: Half relevant, half randomly selected documents. The graph shows the relationship between correlation of the ranked document list (y-axis) and the ESA vector length (x-axis).

around 1/3 of the number of Wikipedia articles in another language version of Wikipedia are connected to their counterpart in the English version, so possibly a significant number of cross-lingual connections might be missing. Which, considering the way the evaluation ground-truth is built (Subsection 2.4.5), can introduce another error.

All of the aforementioned errors surface in the precision-recall metrics. As a result, the link discovery system is undervalued because the results obtained by the skewed evaluation process might substantially diverge from the real performance of the system. As those evaluation metrics are the main direction indicators of the link discovery system research and development, a ground-truth or other method for evaluation of those systems is really needed.

The evaluation for such systems is so far in development and subject of discussion, though we see that a great contribution can be made if the organisers of the NTCIR:CrossLink competition publish the human annotated systems' results. Those could more accurately serve as an objective, although not exhaustive, measure for further comparisons and for development of the cross-lingual link discovery systems.

5.9 Performance Properties

Searching the ESA Vector Index is the most memory exhaustive part of the system. Searching the English Wikipedia ESA Vector Index (3.5M documents) for a keyword takes in average 15 seconds, for a document takes in average 27 seconds [6]. Another performance extensive part is the computation of an ESA vector of a document. In average, the ESA vectors are built out of documents in the speed of 1604 words per second [7]. Therefore, throughout our experiments it proved effective to cache the results of those two methods.

[6] Measured on Intel Core Duo CPU T2350 @ 1.86GHz, 3GB RAM, SSD SATA2.
[7] Measured on Intel Xeon E5620 @ 2.4GHz, 6GB RAM, HDD RAID 1 SATA2.

Chapter 6

Conclusion and Future Work

We have successfully accomplished the assignment. We have investigated and described current automatic link generation approaches. Their summary is given in Chapter 1 and Chapter 2. We have adopted the evaluation metrics and overall nature of the cross-lingual link discovery task from NTCIR (discussed in Chapter 1 and Chapter 5). We designed and implemented different approaches for cross-lingual link discovery, all of which are described and discussed in Chapter 3 and Chapter 4. All methods were evaluated on Wikipedia using the NTCIR, and also also our own data sets. Future directions and conclusion are summarised in this chapter. We have published a paper titled "Using Explicit Semantic Analysis for Cross-Lingual Link Discovery", based on the work described in this thesis [15].

6.1 Contribution

We have designed and implemented several methods for the task of cross-lingual link discovery all of which are based on the Explicit Semantic Analysis, and also its Cross-lingual variation. Effectively, we have prepared a platform for further Cross-lingual Link Discovery research and experiments. The platform can be extended or modified but is a good starting point. We have evaluated the methods on English, Spanish, Chinese and Czech Wikipedia collections and presented results as P-R graphs. Methods based on exploiting the existing link structure of the document collections were clearly superior to the methods based only on document's semantic similarity. We have also analysed certain properties of (Cross-lingual) Explicit Semantic Analysis, particularly its ability to identify the document's counterpart in another language which turns out to be quite good. And also we investigated the influence of dimensionality of ESA vector on semantic similarity rank in a set of documents. That revealed that only about 100 ESA dimensions suffice to prevail the sufficiently correct [1] document order in a document set of semantically similar documents. In addition, we have measured the agreement between link annotations in different language versions of Wikipedia and also between the results produced by our system and the interlinking created by humans. Both agreements were surprisingly low which supports our hypothesis about unsuitability of Wikipedia as an evaluation ground truth for cross-lingual linking systems. Following this, the current cross-lingual linking system's evaluation approach has been summarised and described and arguments about its unsuitability were presented. To facilitate operation of our system we have designed

[1] Correct is the one produced by ordering by similarity using full ESA vectors.

and implemented a vector search engine that facilitates searching for semantically similar documents.

All materials and the source code, along with a brief usage tutorial are available at: http://kmi-crosslink11.googlecode.com.

6.2 Future Directions

For the future we see a chance to analyse the performance of the Cross-lingual Explicit Semantic Analysis when the set of concepts is a parallel corpus. We also believe that the link discovery could be improved by indexing ESA vectors of parts of articles (as opposed to indexing ESA vectors for the whole articles now) for more fine-grained searching for link targets (which could be individual definitions in the articles or its sections). But first, to facilitate this a more sophisticated vector index and search engine needs to be designed and implemented, otherwise this is due to high memory demands impossible. To improve evaluation, a reference corpus should be built and cross-lingually inter-linked. We see Amazon Mechanical Turks, or inventing some kind of language game as ways to facilitate that.

Bibliography

[1] James Allan. Building Hypertext Using Information Retrieval. *Inf. Process. Manage.*, 33:145–159, March 1997.

[2] M. Anderka and B. Stein. The ESA Retrieval Model Revisited. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 670–671. ACM, 2009.

[3] Stephen Dolan. Six Degrees of Wikipedia. [online] http://mu.netsoc.ie/wiki/.

[4] Philipp Dopichaj, Andre Skusa, and Andreas Heß. Stealing Anchors to Link the Wiki. In Geva et al. [7], pages 343–353.

[5] E. Gabrilovich and S. Markovitch. Computing Semantic Relatedness Using Wikipedia-Based Explicit Semantic Analysis. In *Proceedings of the 20th international joint conference on Artifical intelligence*, pages 1606–1611, 2007.

[6] Shlomo Geva. GPX: Ad-Hoc Queries and Automated Link Discovery in the Wikipedia. In Norbert Fuhr, Jaap Kamps, Mounia Lalmas, and Andrew Trotman, editors, *INEX*, Lecture Notes in Computer Science. Springer, 2007.

[7] Shlomo Geva, Jaap Kamps, and Andrew Trotman, editors. *Advances in Focused Retrieval, 7th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2008, Dagstuhl Castle, Germany, December 15-18, 2008. Revised and Selected Papers*, Lecture Notes in Computer Science. Springer, 2009.

[8] Michael Granitzer, Christin Seifert, and Mario Zechner. Context Based Wikipedia Linking. In Geva et al. [7], pages 354–365.

[9] Stephen J. Green. Automated Link Generation: Can We Do Better Than Term Repetition? *Comput. Netw. ISDN Syst.*, 30(1-7):75–84, 1998.

[10] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten. The WEKA Data Mining Software: An Update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

[11] Jiyin He. Link Detection with Wikipedia. In Geva et al. [7], pages 366–373.

[12] Kelly Y. Itakura and Charles L. A. Clarke. University of Waterloo at INEX 2008: Adhoc, Book, and Link-the-Wiki Tracks. In Geva et al. [7], pages 132–139.

[13] Dylan Jenkinson, Kai-Cheung Leung, and Andrew Trotman. Wikisearching and Wikilinking. In Geva et al. [7], pages 374–388.

[14] J. Kamps, J. Pehcevski, G. Kazai, M. Lalmas, and S. Robertson. INEX 2007 Evaluation Measures. *Focused Access to XML Documents*, pages 24–33, 2008.

[15] P. Knoth, L. Zilka, and Z. Zdrahal. Using explicit semantic analysis for cross-lingual link discovery. *Proceedings of the Fifth International Workshop On Cross Lingual Information Access: Computational Linguistics and the Information Need of Multilingual Societies.*, 2011.

[16] Wei Lu, Dan Liu, and Zhenzhen Fu. CSIR at INEX 2008 Link-the-Wiki Track. In Geva et al. [7], pages 389–394.

[17] D. Milne and I.H. Witten. An Open-Source Toolkit for Mining Wikipedia. In *Proc. New Zealand Computer Science Research Student Conf*, volume 9, 2009.

[18] David Milne and Ian H. Witten. Learning to Link with Wikipedia. In James G. Shanahan, Sihem Amer-Yahia, Ioana Manolescu, Yi Zhang, David A. Evans, Aleksander Kolcz, Key-Sun Choi, and Abdur Chowdhury, editors, *CIKM*, pages 509–518. ACM, 2008.

[19] J.L. Myers and A. Well. *Research Design and Statistical Analysis*, volume 1. Lawrence Erlbaum, 2003.

[20] NTCIR organizers. CrossLingual Link Discovery: Evaluation. [online] http://ntcir.nii.ac.jp/CrossLink/Evaluation/, 2011.

[21] D.L. Olson and D. Delen. *Advanced Data Mining Techniques*. Springer Verlag, 2008.

[22] G. Salton and C. Buckley. Term-weighting Approaches in Automatic Text Retrieval. *Information processing & management*, 24(5):513–523, 1988.

[23] P. Sorg and P. Cimiano. Cross-lingual Information Retrieval with Explicit Semantic Analysis. In *Working Notes of the Annual CLEF Meeting*. Citeseer, 2008.

[24] P. Sorg and P. Cimiano. An Experimental Comparison of Explicit Semantic Analysis Implementations for Cross-language Retrieval. *Natural Language Processing and Information Systems*, pages 36–48, 2010.

[25] Jihong Zeng and Peter A. Bloniarz. From Keywords to Links: an Automatic Approach. *Information Technology: Coding and Computing, International Conference on*, 1:283, 2004.

[26] Torsten Zesch, Christof Mueller, and Iryna Gurevych. Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary. In *Proceedings of the Conference on Language Resources and Evaluation (LREC)*, 2008.

[27] Junte Zhang and Jaap Kamps. A content-based link detection approach using the vector space model. In Geva et al. [7], pages 395–400.