Replicating an Experiment in Cross-lingual Information Retrieval with Explicit Semantic Analysis

Marco Jungwirth (✉) and Allan Hanbury[0000−0002−7149−5843]

Institute of Information Systems Engineering, TU Wien, Favoritenstraße 9-11/194, 1040 Vienna, Austria
e1326215@student.tuwien.ac.at, allan.hanbury@tuwien.ac.at

Abstract. We have participated in the Replicability Track of the CENTRE@CLEF 2018 conference [1–4]. This paper reintroduces Explicit Semantic Analysis (ESA) and its extension for cross-lingual document retrieval tasks, called Cross-lingual Explicit Semantic Analysis (CL-ESA), for the first time introduced by Sorg and Cimiano in 2008. The goal is to replicate an experiment from Sorg and Cimiano, who participated in the CLEF conference in 2008 and report on the results as well as to point out mistakes and problems along the way. This work should be read in conjunction with the original work done by Sorg and Cimiano [7].

Keywords: replicability · Explicit Semantic Analysis · cross-lingual.

CEUR-WS.org/Vol-2125/paper_208.pdf

1 Introduction

The goal of this paper is to replicate an experiment and its results as reported in the paper *Cross-lingual Information Retrieval with Explicit Semantic Analysis* by Philipp Sorg and Philipp Cimiano. This paper is structured as follows: in Section 2, we concisely introduce Explicit Semantic Analysis [5]. Section 3 describes the extension of this approach for cross-lingual document retrieval using cross language links from Wikipedia. In Section 4 the experimental setup and the differences to the original experiment are described in detail. Finally, in Section 5 we present the results of the replicated experiment.

2 Explicit Semantic Analysis

Explicit Semantic Analysis (ESA) uses chosen external categories to represent a given text t. We will introduce the main ideas of a Wikipedia based approach, so the reader is able to understand the notions in the implementation section. A more detailed description can be found either in the paper by Sorg and Cimiano [7] or by Markovitch and Gabrilovich [5], who introduced this Wikipedia-based approach and the more general theory behind it. The main idea in Wikipedia-based Explicit Semantic Analysis is to map a text t into

a high-dimensional real-valued vector space. Given a set of Wikipedia articles $W_k = \{a_1, \ldots, a_n\}$ in language $\mathcal{L}_k$, each article a_i is an external category and corresponds to a dimension in the vector space. The following function describes this mapping:

$$\Phi_k : T \to \mathbb{R}^{|W_k|}$$

$$\Phi_k(t) := \langle v_1, \ldots, v_{|W_k|} \rangle$$

where $|W_k|$ is the number of articles in W_k. Each v_i is computed by summing up the results of a function as, which defines the *strength of association* between a Wikipedia article a_i and a word w_j, for each word in a given text $t = \langle w_1, \ldots, w_l \rangle$. In this regard $\Phi_k(t)$ is called ESA-vector and expresses the strength of association of a given text t with each article a_i in W_k.

$$v_i := \sum_{w_j \in t} as(w_j, a_i)$$

There are many different ways to define the as function, e.g. here we use a $tf.idf$ function, based on a Bag-of-Words model of the Wikipedia articles. In this sense, Explicit Semantic Analysis is very flexible, as it can be adapted to different tasks and contexts, simply by choosing a different as function.

$$as(w_j, a_i) = tf.idf_{a_i}(w_j)$$

The function we used for the experiment is described in Section 4. Computing the ESA-vector, means we compute the strength of association for each Wikipedia article a_i, hence after sorting the ESA-vector by value, it corresponds to a ranking of Wikipedia articles according to relevance for a given text t. Essentially, Explicit Semantic Analysis transforms a given text t into a vector representation according to external categories. This means one can simply assess the similarity between two arbitrary texts t_1 and t_2 by computing their ESA-vectors and for example using the standard cosine similarity to compare the vectors. This is another reason for which Explicit Semantic Analysis is very flexible, as it can be used on arbitrary texts — we can simply adapt it to different tasks, among other things a retrieval task (query and document) or a clustering task (two documents).

3 Cross Lingual Explicit Semantic Analysis

Cross Lingual Explicit Semantic Analysis (CL-ESA) is an extension to ESA, which can handle multi-lingual retrieval tasks. This approach uses the fact that Wikipedia articles are linked across different languages. Therefore one can assume that there exists a mapping function $m_{i \to j}$, which maps an article a_i from Wikipedia W_i to its corresponding article in Wikipedia W_j. Suppose there are n languages $L_1, \ldots, L_n$. Transforming a given text t from language L_i to language

L_j is as simple as transforming $\Phi_i(t)$ to $\Phi_j(t)$ using a map, which is defined over the cross language links in Wikipedia W_i. Since we consider n languages, we define an n^2 mapping function of the type:

$$\Psi_{i \to j} : \mathbb{R}^{|W_i|} \to \mathbb{R}^{|W_j|}$$

This mapping is computed as follows:

$$\Psi_{i \to j} \langle v_1, \ldots, v_{|W_i|} \rangle = \langle v'_1, \ldots, v'_{|W_j|} \rangle$$

where

$$v'_p = \sum_{q \in \{q* | m_i \to m_j(a_{q*}) = a_p\}} v_q$$

with $1 \leq p \leq |W_i|, 1 \leq q \leq |W_j|$. Given a text t in language $\mathcal{L}_i$, obtaining an ESA-vector from Wikipedia W_j, is as simple as computing $\Psi_{i \to j}(\Phi_i(t))$. Using the above setting we can now define the cosine between a query q_i in language $\mathcal{L}_i$ and a document d_j in language $\mathcal{L}_j$ in a straightforward manner as follows:

$$\cos(q_i, d_j) := \cos(\Phi_i(q_i), \Psi_{j \to i}(\Phi_j(d_j)))$$

Now we obtained a uniform approach across multiple languages. Nevertheless, it is important to note that CL-ESA works under the assumption that the language of the document is known.

4 Implementation

In this section we will describe the implementation details used for the experiment. Unfortunately the overall setup differs from the original experiment, because we were not able to obtain database dumps from 2008. Instead we downloaded static HTML dumps[1] in English, German and French from the year 2008 and extracted them on the disk [1–4].

4.1 Preprocessing of the documents

For the actual indexing step we used the same methods as Sorg and Cimiano, namely a standard white space tokenizer, standard stop word lists for English, German and French and a Snowball[2] Stemmer for English, German and French respectively.

4.2 ESA Implementation

In this section we will give a detailed description of our implementation and preprocessing of the documents and Wikipedia articles. Moreover we will compare our implementation with the implementation described in the original paper by Sorg and Cimiano.

[1] `https://dumps.wikimedia.org/other/static_html_dumps/2008-06/`
[2] Snowball Stemmers are included in Lucene

Wikipedia Article Preprocessing After the extraction we realized quickly, that the static HTML dump has much more pages than just the article pages we wanted to index. Hence the first step for us was to write a python script which ignored Wikipedia specific pages. Fortunately these pages were easy to locate as their purpose was encoded in the filename of the page, namely pages which started with Category, Image, Portal, Help, Template, User or Wikipedia[3] were ignored and the corresponding discussion pages[4] were ignored as well. The python script simply changed the file extension from *.html* to *.ign*, which stands for ignore. Moreover every page with a filesize of less than 1KB was ignored, due to the fact that those files were redirect pages. The HTML markup for an actual article would already exceed this limit, hence there was no room for erroneously ignoring an actual article. Now the indexer would only index pages whose file extension is not *.ign*.

The processing of the Wikipedia articles is vastly different to Sorg's approach, due to using a different source. First we needed to extract the relevant text from the HTML mark up, using a library called JSoup[5]. This library empowered us to query the HTML markup using CSS-style queries. With this approach we selected the div-element with *id* equal to *content*. Then we removed the table of contents which is a table tag with *id* equal to *toc*. Moreover we removed any category links, which is a div-element with *id* equal to *catlinks* and we removed all edit sections, which were span tags with *class* equal to *editsection*. After these steps we simply selected the text between all remaining tags and used this as a document for the index. We randomly sampled about 30 articles and looked at the result of this process to convince ourselves that there are no more unnecessary texts which might skew the index [1–4]. Nevertheless there were cases where this approach threw exceptions and we printed each of them into a logfile. However, the number was less than 200, when compared to the number of documents in the corpus which are more than half a million documents, we decided that it was not worth it to investigate those articles further at this stage.

For indexing the documents using Lucene[6], it was not necessary to use the WikipediaAnalyzer, because our text was just plain text without Wiki markup. Therefore we used the same methods as in the preprocessing of the documents. Sorg and Cimiano mention two kinds of restrictions on the article selection, "Then all articles with less than 100 words or less than 5 incoming pagelinks were discarded." We implemented them adapted to the static HTML dump. The length is checked before adding a document to the index by counting the tokens generated from Lucene. The incoming page links were more complex to obtain. We generated a page link map by parsing all Wikipedia articles and counting the number of *a href* tags in the div-element with *id* equal to *content*. The

[3] The prefixes were in the same language as the corresponding Wikipedia, e.g. Benutzer for User in the German Wikipedia

[4] for each prefix there was a discussion page encoded as <prefix>_talk

[5] `https://jsoup.org/`

[6] `https://lucene.apache.org/core/`

link[7] in these tags is a path to the corresponding article in the filesystem and we used them as keys for the aforementioned page link map and counted the occurrences. Then instead of parsing the whole Wikipedia directory with the indexer, we simply looped over all keys in the map and only parsed the articles with 5 or more page links. Unfortunately, there are no exact document counts from Sorg and Cimiano after applying these restrictions. Nevertheless, they reported document counts after restricting the documents to "at least a language link to one of the two other languages we consider [...] we used 536.896 English, 390.027 German and 362.972 French articles for the ESA indexing" [7] We ended up with more documents for the ESA indexing than Sorg and Cimiano. The English Wikipedia index consists of 1.517.398 documents, the German Wikipedia index consists of 520.433 documents and the French Wikipedia index consists of 431.245 documents. Considering the additional restriction Sorg and Cimiano applied to obtain their counts, we think that the discrepancy between our numbers is reasonable [1–4]. The English Wikipedia is much larger than the German or French Wikipedia. Therefore there are a lot more pages which do not have a link to German or French. The smaller discrepancy in the French and German Wikipedia are probably pages unique to their cultural heritage and therefore are not likely to have an English equivalent. Nevertheless, we did not check any of the aforementioned reasons, because the additional restriction is unrelated to replicating the experiment at hand. That being said, having a vastly different preprocessing and wikipedia source is probably a solid reason, why we were not able to obtain results similar to Sorg and Cimiano.

ESA Vector Computation The computation of the ESA vector uses an inverted index of the selected Wikipedia articles. Each document will be queried against this index and the retrieved articles will be used to build the ESA vector. Similar to Sorg and Cimiano, we used Lucene for indexing and the association strength was implemented using a customized Lucene similarity function. The function takes a text $t = \langle w_1, \ldots, w_l \rangle$ and a Wikipedia article a_i of Wikipedia corpus $|W|$ and computes the following function:

$$as_R(t, a_i) = (C_t)\sqrt{|a_i|}^{-1} \sum_{w_j \in t} tf_{a_i}(w_j) idf(w_j)$$

with

$$C_t = \frac{1}{\sqrt{\sum_{w_j \in t} idf(w_j)}}$$

$$tf_{a_i}(w_i) = \sqrt{\#\text{occurrences of} w_i \text{in} a_i}$$

$$idf(w_j) = 1 + \log \frac{|W| + 1}{\#\text{articles containing} w_j}$$

[7] removed anchorpoints

The following *idf* is described in the original paper by Sorg and Cimiano:

$$idf(w_j) = 1 + \log \frac{\#\text{articles containing} w_j}{|W| + 1}$$

First we need to point out an error in defining the *idf* function the way it was defined by Sorg and Cimiano [1–4]. The result of the idf function would be negative because the fraction is definitely less than 1 and taking the log of a value less than 1 yields a negative result. We realized, that this is probably an error because in the literature (e.g. [6]), variations of the *idf* are defined differently and result in a positive value greater than 1. Therefore we swapped the numerator with the denominator and used this variant for our experiment.

Multi-lingual Mapping Similar to Sorg and Cimiano some preprocessing was needed to obtain the multi-lingual mapping. Due to using the static HTML dump instead of a database dump, the cross language links were embedded in the HTML and pointed to the actual filename on the filesystem. The replicated experiment only involved English topic titles, hence we only computed the mapping from German to English and from French to English. A normalization of the page title was not needed, because we computed the mapping using the document ids from the index, which correspond to the index of the dimension in the ESA-vector. Nevertheless we needed to deal with redirect pages [1–4], therefore we used the following steps to compute the mapping:

1. For each document in the German (resp. French) index
2. Find the file in the file system and check[8] if a link to English is available
3. If an English link is available find the file in the file system.
4. Recursively determine if the file is a redirect page until the actual document is reached.
5. Look up the document id in the English index and add it to the mapping.

Similar to Sorg and Cimiano we summed up the scores, in case of multiple language links pointed to the same article in the English Wikipedia.

4.3 Language Identification

The computation of the ESA vector is based on the assumption that we know the language in which the document is written. Unfortunately this is not always the case. Even in the TEL German dataset used for the replication, there are records without any knowledge about the language. Hence Sorg and Cimiano presented the following function to determine the language of a document t:

$$lang(t) := \max_{L_k \in \{L_1, \ldots, L_n\}} \frac{minDim(\Phi_k(t))}{maxDim(\Phi_k(t))}$$

[8] We selected the div-element with *id* equal to *p-lang* with JSoup

where "$minDim(\Phi_k(t))$ returns the value of the lowest dimension in vector Φ_k and $maxDim(\Phi_k(t))$ returns the highest correspondingly." [7] To us this description of lowest and highest dimension of a vector does not make sense. We thought of multiple possibilities to interpret it, e.g. index of the dimension with the lowest and highest value of the vector or the actual minimal and maximal values of the vector. However, none of this made sense. Fortunately, Sorg and Cimiano give an intuition about their heuristic as "The intuition behind this heuristic is that a small difference between the values of the lowest and highest dimension, which is computed by the share of these values, means that the document matches good to many Wikipedia articles and it can therefore be assumed that the document is of the same language as the used Wikipedia articles. Comparing a document to Wikipedia articles in another language, there will be some matches but the value of lowest dimension will most probably be very small." Following this intuition lead us to interpret it as follows:

$$lang(t) := \max_{L_k \in \{L_1,\ldots,L_n\}} \frac{\#\text{non zero elements of } \Phi_k(t)}{|W_k|}$$

where $|W_k|$ is the number of articles used from Wikipedia in language k [1–4]. This way we get the percentage of Wikipedia articles a document t is matched to. Then the language, in which the document should be written in, is the language of the Wikipedia base with the highest percentual article match. We have implemented our interpretation of the language identification, however when trying to identify the language of records without language tag, the run would have taken too long and we would not have been able to submit our results in time. Therefore we chose to try and match a document without language tag with the German Wikipedia by default.

4.4 Retrieval

The retrieval algorithm we used, presented in Algorithm 1, is generally the same as Sorg and Cimiano presented in their paper. The only change here is that our language identification solely relies on language tags in the records to identify the correct language of the document.

5 Results

In this section we present additional information about the dataset, its language distribution, additional settings of the experiment and the results.

5.1 Dataset

The TEL German dataset has in total 869353 records in over 100 different languages. The data can be split in records with a language tag, which is about 90%, and without a language tag, which is the rest. German, English and French are the main languages of those records with language tag and make up about

Input: Topics T, Language k, Documents D
for $t \in T$ **do**
  $\boldsymbol{t} = \Phi_k(t)$;
end
for $d \in D$ **do**
  $l := lang(d)$;
  $\boldsymbol{d} = \Psi_{l \to k} \Phi_l(d)$;
  for $t \in T$ **do**
    $score\ [t, d] = \cos(\boldsymbol{t}, \boldsymbol{d})$;
  end
end

Algorithm 1: Retrieval-Algorithm

88%. In our experiment we only use the title information to build queries for the index and according to Sorg and Cimiano "The title of the record is the only content information that is available for all records". We cannot confirm this statement, because we have found that about 4,5% of the records in the dataset do not contain title information [1–4]. In our experiment we simply ignore the records that do not contain this information.

5.2 CLEF Replicability Experiment

The objective of this experiment is to query the 50 given topics in English on the multi-lingual TEL German dataset [1–4]. The topics consist of a title and a short description to build a query. Sorg and Cimiano do not mention what they used to build the query. We chose to use only the title as a query. As for the ESA-vector length k we ended up trying two different settings. Sorg and Cimiano used $k = 10.000$ for topics and $k = 1.000$ for the records. We ran one experiment with the same settings and additionaly we ran another experiment with values of a magnitude smaller, namely for the topics we used $k = 1.000$ and for the records we used $k = 100$. The result obtained by Sorg and Cimiano was a mean average precision (MAP) of 6,7% [7]. Unfortunately, they did not explicitly mention a requirement for a record to count as a relevant document for a certain topic. Therefore we assumed, that every score greater than zero is a relevant document. This means that as soon as there is one overlapping dimension in the ESA-vectors of a record and a topic it would yield a relevant document. This assumption lead us to the problem, that the full list of relevant documents, obtained from the experiment with the larger ESA-vector length, matched more than two thirds of all the records in the dataset to nearly every topic. Looking at the list we figured out, that the score of the higher ranking documents decreased at a faster pace and the relevant documents after rank 1000 decreased in a much slower pace and yielded only a small fraction of the score in comparison to the higher ranking documents. We concluded, that it would be meaningful to cut off the results at a certain rank and look at the MAP of the reduced lists, because the relevant documents with a very low score might just have been accidently connected by a single Wikipedia article, which does not necessarily convey a

semantical connection between a record and a topic. We used the top 10, the top 100 and the top 1000 results for each topic, and we calculated the MAP using trec_eval[9]. The results are shown in Table 1. After comparing our results with the 6,7% obtained by Sorg and Cimiano, we conclude, that we were not able to reproduce the result.

Table 1. Mean Average Precisions of the experiments (values in %)

ESA length (Topic/Record)	Top 10	Top 100	Top 1.000
1.000/100	0,65*	0,1	0,01
10.000/1.000	0,63*	0,09*	0,01

* submitted as run to the replicability track

6 Conclusion

In this paper, we described the CL-ESA approach presented by Sorg and Cimiano and we attempted to replicate an experiment, submitted to the CLEF conference in the year 2008. In the end we were not able to reproduce the result. Most parts of the experimental setup were replicated accurately, but especially the index might be very different in comparison with the index of the original experiment, because we were not able to obtain a Wikipedia database dump from 2008 and therefore worked with a static HTML dump. Since the index is at the core of the experiment, it can lead to subsequent differences in every other part. Other than that, some missing details, e.g. the way the query is built from the topics and a detailed explanation about what fields from the records of the dataset were used, make it hard to replicate the experiment in a more detailed manner. Moreover we ran into some problems based on our own assumptions. We are refering to the fact that the full result list of the experiment with the bigger ESA-vector lengths matched two thirds of all the records in the dataset to almost every topic. We think, that the cause of this problem lies in the fact, that while there are Wikipedia articles, which might very accurately describe a semantical category, there are certainly some articles, which have the opposite effect. To give a short example, suppose an article of a famous actress will have a lot of different words with different semantical meanings on her page (e.g. overview of her career and life), while also having acted in some horror movies. This article would then match any kind of record of the dataset, which somehow was able to obtain a positive score through words which are not semantically connected to horror movies, to the topic horror movies, even though there is probably no semantic connection whatsoever. Therefore we think, that for Wikipedia-based CL-ESA to yield better results it is essential to have a good article selection or to introduce certain restrictions on what ends up being a relevant document for a certain topic, e.g. at least 10 dimensions need to overlap in the ESA-vectors.

[9] `https://github.com/usnistgov/trec_eval`

References

1. Bellot, P., Trabelsi, C., Mothe, J., Murtagh, F., Nie, J.Y., Soulier, L., SanJuan, E., Cappellato, L., Ferro, N. (eds.): Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Nineth International Conference of the CLEF Association (CLEF 2018). Lecture Notes in Computer Science (LNCS), Springer, Heidelberg, Germany (2018)
2. Cappellato, L., Ferro, N., Nie, J.Y., Soulier, L. (eds.): CLEF 2018 Working Notes. CEUR Workshop Proceedings (CEUR-WS.org), ISSN 1613-0073 (2018)
3. Ferro, N., Maistro, M., Sakai, T., Soboroff, I.: CENTRE@CLEF2018: Overview of the Replicability Task. In: Cappellato et al. [2]
4. Ferro, N., Maistro, M., Sakai, T., Soboroff, I.: Overview of CENTRE@CLEF 2018: a First Tale in the Systematic Reproducibility Realm. In: Bellot et al. [1]
5. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using wikipedia-based explicit semantic analysis. In: Proceedings of The Twentieth International Joint Conference for Artificial Intelligence. pp. 1606–1611. Hyderabad, India (2007), `http://www.cs.technion.ac.il/~shaulm/papers/pdf/Gabrilovich-Markovitch-ijcai2007.pdf`
6. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA (2008)
7. Sorg, P., Cimiano, P.: Cross-lingual Information Retrieval with Explicit Semantic Analysis. In: Working Notes for the CLEF 2008 Workshop (2008)