AUT Document Alignment Framework for BUCC Workshop Shared Task

Atefeh Zafarian, Amirpouya Aghasadeghi, Fatemeh Azadi, Sonia Ghiasifard, Zeinab Alipanahloo, Somayeh Bakhshaei, Seyed Mohammad Mohammadzadeh Ziabary
Human Language Technology Lab
Computer Engineering Department
Amirkabir University of Technology, Tehran, Iran
{atefeh.zafarian, aghasadeghi, ft.azadi, s.ghiasi, apanahloo, bakhshaei, mehran.m}@aut.ac.ir

Abstract

This paper presents a framework for aligning comparable documents collection. Our feature based model is able to consider different characteristics of documents for evaluating their similarities. The model uses the content of documents while no link, special tag or Metadata are available. And also we apply a filtering mechanism which made our model to be properly applicable for a large collection of data. According to the results, our model is able to recognize related documents in the target language with recall of 45.67% for the 1-best and 62% for the 5-best.

1 Introduction

Comparable corpora (CC) are collections of similar documents with different levels of comparability (Fung and Cheung, 2004). There are useful resources for most of the Natural Language Processing (NLP) or Information Retrieval (IR) tasks such as cross-lingual text mining (Tang et al., 2011), bilingual lexicon extraction (Li and Gaussier, 2010), cross-lingual information retrieval (Knoth et al., 2011) and machine translation (Smith et al., 2010; Delpech, 2011) etc.

The sub-fields of NLP are related to solving human language tasks that are mostly hard problems such as Language Understanding (Winograd, 1972), Machine Translation etc. The modern algorithms of NLP sub-fields are based on machine learning and statistical approaches. Most of the developed systems of these fields require large amounts of parallel corpora, as a result the limitation in success of such tasks is the lack of parallel corpora. In recent researches, it is proven that Comparable Corpora can be a valuable alternative to rare parallel corpora.

Information Retrieval (IR) is "the act of finding materials, usually documents of an unstructured form that satisfies an information need within large collections stored in computers" (Manning et al., 2008). IR is not limited to monolingual documents if the task is related to mapping bilingual or multilingual documents; a new area of IR will be introduced: Cross/Multilingual IR. The idea of Cross-Lingual IR (CLIR) is to retrieve documents in a language different from the language of input text (Oard, 1998). The input text may be either a query or a document which categorizes the field to document based or query based approaches. CLIR is a way of expanding input queries to other languages. This is a useful approach in search engines that enables users to formulate queries in their preferred languages and retrieve relevant documents in whatever language they are written. For this purpose instead of parallel corpora for translating input queries, using comparable corpora might be helpful. However, document based CLIR can be used for producing comparable corpora. The related works will be reviewed in section 2.

Our Model is a framework consists of different modules. Each module considers disparate features for matching each source document with the target documents, so we called this a feature-based model. The pipeline of the modules in our model is shown in Figure 1.

We assume two similar documents contain same sets of names which occur in the same order. Name Module of our model is responsible for checking this structure. In addition, translation of similar texts in the source and target languages must be similar, so we use SMT system as another module in the model. We also assume similar documents converted to vector representations using neural networks will have shorter Euclidean distance between each other. This characteristic of similar documents is considered

Proceedings of the Eighth Workshop on Building and Using Comparable Corpora, pages 79–87,
Beijing, China, July 30, 2015.

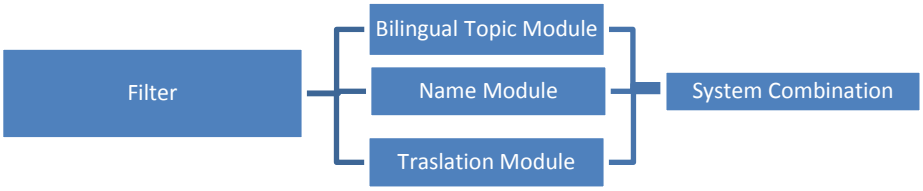


Figure 1. The structure of the pipeline model.

in Document-to-Vector module. To recognize similar conceptual structures in documents we use bilingual topic models.

The problem is aligning source documents to target documents, but because of the amount of data, in addition to similarity concerns, our model is faced to the very large corpora problem. It is not possible to evaluate each pair of documents in this big space, so we used some of our modules as a filter for decreasing the target space. From the framework's modules, we choose ones with higher Recall to be sure that our filter is able to recognize and remove wrong samples. Another factor for selecting the filtering modules is the execution speed. Low speed modules are not proper in the model's pipeline.

2 Related Works

Common methods for comparing documents are by extracting features from texts; namely compare documents through the most frequent words (Kilgarriff, 2001).

In multilingual context, some approaches translate features and compare documents using differences in the frequencies of the translations of the keywords, namely using cosine similarity measure between the feature vectors (Su and Babych, 2012).

A successful approach is the Cross Language Character N-Gram (CL-CNG) model (Mcnamee and Mayfield, 2004) that uses character n-grams and is based on the syntax of documents, found remarkable performance for languages with syntactic similarities.

A primary approach for aligning comparable document corpora is based on statistical machine translation technology such as CL-ASA (Barrón-Cedeno et al., 2008), that uses a combination of a translation model and a length model for measuring similarity between source and target documents. The translation model shows that how likely the source document is a translation of the target document and length model measures the similarity of those two documents with the length attribute. It is expected for a pair of translated documents to have closely related lengths.

A common language independent approach for representing documents is based on vector representation. Representing documents in a collection as bag off words is called Vector Space Model. Each component of the vector shows the importance of that term in the document. In large document collections, document vectors have high dimensions. For this reason, some approaches using linear projection, a map from the high dimension to a low-dimensional vector space. Early approaches for linear projection are LSA (Deerwester et al., 1990) and LDA (Blei et al., 2003).

Cross-language latent semantic indexing (CL-LSI) (Dumais et al., 1997) is based on LSA used for multiple languages by reducing the dimensionality of a matrix which rows are obtained by concatenating comparable documents from different languages. Another projection model, Latent Dirichlet Allocation (LDA) is based on the extraction of generative models from documents. Polylingual Topic Models (Mimno et al., 2009) are multilingual versions of LDA.

Cross Language Explicit Semantic Analysis (CL-ESA) is the other model in vector context approach (Potthast et al., 2008) that uses comparable Wikipedia corpora. Each document is represented by a concept vector, where each dimension is the similarity of the document to one of the Wikipedia documents in the corpus.

New approaches for comparable document retrieval task and for measuring documents similarities are knowledge-based; despite previous works that were supervised. Knowledge-based Document Similarity (KBSim) (Franco-Salvador et al., 2008) is one of the most recent of them. It turns source and target documents to knowledge graphs using a Multilingual Semantic Network (MSN) such as Babelnet (Navigli and Ponzetto, 2012) then compares two graphs using KBSim.

3 Model description

The framework of our model is constructed on 4 modules: Doc2Vec, Name, Topic Model and SMT. These modules evaluate the similarity of each pair of documents considering different characteristics of a document pair.

According to the framework of our model (Figure 1), the first step contains filters for reducing the size of the target space. Two filters are used serially based on Doc2Vec and Name modules for this purpose. In the following subsections, we explain each of the modules used in our framework in more details.

3.1 Document-to-Vector Module (Doc2Vec)

Recent works in learning vector representation of words using neural networks (Mikolov et al., 2013), show that these models can capture great details about semantics and syntactic relationships and patterns between similar words, which many of those patterns can be obtained from simple linear transitions. For example, it has been shown that the result of a vector calculation vec("Madrid") - vec("Spain") + vec("France") is closer to vec("Paris") than to any other word vector (Mikolov et al., 2013).

Another great properties of these models is that if they trained on comparable corpora in different languages, by using simple transformation matrix, vectors from source language can be projected to the target space and be used to build a larger dictionary (Mikolov et al., 2013). Such transformation matrices can be obtained from a few thousand aligned words from the source and target side.

Models mentioned above can only work on fixed length text inputs such as words or short phrases, but many tasks in NLP need variable input length. A new extension of these models is Paragraph Vector (Le and Mikolov, 2014) which can convert any variable length input from sentence to document, to a fixed length vector output. Since the Paragraph Vector model training is similar to Word to Vector model, they share many properties and this new model also captures the relationship between similar words and sentences. The previous works on this model show the state of the art results in the field of sentiment analysis and document classification.

As far as we know there has been no previous use of Paragraph Vector model for bilingual and multilingual tasks. Since the Paragraph Vector model is based on Word to Vector model, we get to this conclusion that by using the same method mentioned in (Mikolov et al., 2013) we can build a bilingual Paragraph model to align source and target Documents.

The transformation matrix can be found by solving following optimization problem. In equation (1) x_i is the vector representation of i_{th} source document and z_i is its paired document vector representation in the target space.

$$\min_w \left(\sum_{i=1}^{n} \| W x_i - z_i \| \right) \quad (1)$$

W can be found by any optimization method, but we solved it with a stochastic gradient descent approach. By computing $z = Wx$ any source vector will be projected to the target space, then we can search closest neighbors of z in target vectors to find our answers.

Training this model and transformation matrix is relatively fast in comparing with our other modules. Even though this method precision is low, it can discriminate related and unrelated documents from each other very well. Since generating closest neighbors list in this method is simple, this module is used for filtering the target search space for our slower modules such as topic models and machine translation.

3.2 Bilingual Topic Model Module (BiTM)

The basic idea behind topic models is that documents are mixtures of topics, where a topic is a probability distribution over words (Blei et al., 2003). Topic models have a major benefit; they don't need documents to be sentence-aligned, so it will be a good choice for finding comparable corpora. To model bilingual topics, we used an extension of latent Dirichlet allocation (LDA) called Polylingual Topic Model (Mimno et al., 2009). We consider each document as a bag of words, this approach consists of three main steps, first step is creating sets of topics for both sides (source and target languages) then calculating probability of each topic in each document and finally, finding documents similarities.

Figure 2 shows graphical model of polylingual topic model, where α and β are the hyperparameters on the Dirichlet priors for topic distributions θ and the topic-word distributions φ respectively. This model actually finds and aligns topics of different languages.

Now that the topic distributions of target and source languages are created, we use these topics to find topics probabilities over each document using Gibbs sampling.

Accordingly, each document is converted to a T dimensional vector $v = [p_1, p_2, \dots, p_T]$, where p_1 is the probability of assigning topic one to this document and T is the number of topics. To find similar documents in two languages we used a well-known method called cosine similarity. In our case, two vectors (from source and target language) are compared, using cosine similarity as bellow:

$$Sim(v', v) = \frac{\sum_{i=1}^{n} v'_i \times v_i}{\sqrt{\sum_{i=1}^{n} (v'_i)^2} \times \sqrt{\sum_{i=1}^{n} (v_i)^2}} \quad (2)$$

Where v' and v are respectively documents of source and target language. The result is a number between 0 and 1, while the similarity of 1, means the documents are completely similar.

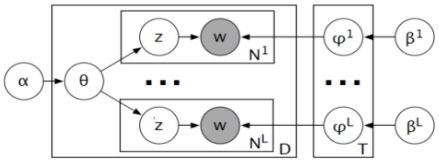


Figure 2. Graphical model of topic model (Mimno et al., 2009)

3.3 Names Module

Named entities play an important role in Cross-Lingual Information Retrieval (CLIR). Comparable documents generally share many named entities (NE) (Gupta and Bandyopadhyay, 2013), in this section, we make a Name model for checking the effectiveness of NEs in aligning comparable documents. In this model, we extract NEs from documents and classify into three types: location, person and organization, then compute document similarity based on the similarity of names that have the same type. As respects NEs usually are phonetically transliterated, we consider the phonetic similarity of the two words as similarity criteria. Our Name model contains two sections:

A. Named Entities Recognition: we apply a CRF-based supervised classifier as NER model.

B. Computing Phonetic Similarity: The main bottleneck in computing phonetic similarity is the lack of availability of transliteration training data so we propose a solution for solving this problem. Our proposed method includes 3 following steps:

3.3.1 Transliteration Mining

In this step, we use an unsupervised transliteration mining model for extracting transliterated names from parallel corpus that is described in (Durrani et al., 2014) and apply this on the Europarl parallel corpus and extract a transliterated bilingual German-English dictionary that we called ENTD (Europarl Transliterated Names Dictionary).

3.3.2 Mapping Table

In this step, we extract high-probability transliterated names of ENTD and apply an iterative alignment model on this for generating a table of characters that are aligned with high probability in source and target languages. This method is similar to the method described in (Mousavi Nejad and Khadivi, 2011). The alignment model is a Levenshtein distance based on the mapping table. In each iteration of the model, the characters with high alignment probabilities added to the mapping table and the algorithm is repeated until no change in the mapping happens anymore.

3.3.3 Compute Phonetic distance

In this step, we compute the phonetic distance between name entities in comparable training documents using a recursive function based on the mapping table. This method is similar to the method described in (Mehdizadeh Seraj et al., 2014). For measuring the distance between an English character in position i and a German character in position j. We will use the recursion definition, according to the following equation. In this definition, e and g are English and German words respectively.

$$d(i, j) = \min \begin{pmatrix} d(i\text{-}1, j) + \left(1 - p_{remove(e_i)}\right), \\ d(i, j\text{-}1) + \left(1 - p_{remove(g_i)}\right), \\ d(i\text{-}1, j\text{-}1) + \left(1 - p_{replace(e_i, g_i)}\right) \end{pmatrix} \quad (3)$$

Where p_{remove} and $p_{replace}$ are obtained from the mapping table. Finally, we generate a transliterated bilingual German-English dictionary of transliterated names that have a low phonetic distance, named BTND (BUCC Transliterated Names Dictionary).

3.3.4 Compute Similarity of Documents in Test Time

Computing the phonetic similarity by a recursive function takes a lot of time and it is not efficient for test time, so we use the bilingual dictionaries generated in the previous sections. When there is enough time, we can use the method described in

section 3.3.3 that is a language-independent method. In this state, we divide source-target names in each of the two documents in 3 states: 1. The named entities of the same type that have same letter form. 2. The named entities of the same type that exist in ETND. 3. The named entities of the same type that exist in BTND.
We search each source-target name in state 3 only if it doesn't exist in state 1 and 2, and search it in state 2 only if it doesn't exist in state 1. We also extract URLs from documents and consider these as state 4:

4. The same URLs in two documents.

Finally, we define a score function for computing document similarity between a German document G and an English document E as follow:

$$score_{G,E} = \frac{w_1 s_1 + w_2 s_2 + w_3 s_3 + w_4 s_4}{C_{NE} + C_{url}} \quad (4)$$

Where w_1 is the weight of state 1 and s_1 is the number of common NEs in documents G and E. C_{NE} and C_{url} are the number of NEs and URLs in German documents. For estimating the weight of each state, we apply our models on a development set and increase the impact of phonetic dictionaries ETND and BTND by filtering the pairs of names with low alignment probabilities. We compute the thresholds by testing on the development set.

3.4 SMT Module

When two documents in two different languages are similar, the translation of the first document to the other's language should make a similar document to the second one. That's why we use the statistical machine translation (Brown et al., 1993) as another module for measuring the document similarities.
In this module, we first train a phrase-based SMT system on a sentence-aligned parallel corpus (Zens et al., 2002; Koehn et al., 2003). Then we translate each source document with the trained SMT system, and in the next step, for each source document we calculate similarity scores by comparing its translation to the list of filtered target documents that were produced by the former modules.
For the similarity scores, we use two well-known translation evaluation metrics. The first metric is BLEU, which is computed by comparing the system output against the reference translation (Papineni et al., 2002). Given the precision p_n of n-grams of size up to N (usually N = 4), the length of the translation output in words (c), and the length of the reference translation in words (r), the BLEU metric will be computed as follows,

$$BLEU = BP.\exp(\sum_{n=1}^{4} \log p_n) \quad (5)$$

$$BP = \min(1, e^{1-r/c}) \quad (6)$$

Here the translation output is our SMT system's output for the source document and the reference translation is a target document. As these two documents are not necessarily sentence-aligned we concatenate each of them to make one sentence documents. As we know, one of the BLEU metric's shortcomings is that it was designed for corpus level and might not perform well on single sentences, since the 4-gram precision could be often zero and it makes the whole BLEU score to be zero.
So as BLEU might perform badly in some cases, we also used another metric called Position-independent word Error Rate (PER) (Tillmann et al., 1997). This metric measures a position-independent Levenshtein distance (bag-of-word based distance) between the translation output and reference. The resulting number is then divided by the number of words in the reference.
The reason that we used this instead of other error rates such as WER (Nießen et al., 2000) and TER (Snover et al., 2006) is that it completely neglects word orders. As in our task, sentences in two similar documents might be displaced and we don't want this displacement to influence our similarity score, PER is more reasonable to use.
As the BLEU score contains higher order n-grams, it also considers correct phrases instead of just words in PER, and so it has a higher recall in our experiments (shown in section 4). But as PER might help for the cases that BLUE is not working well, we use both of these scores for our final system.

3.5 System Combination

In our model first the big space of English documents is filtered with high-speed modules. Then for each pair of the documents in this filtered space we compute the value of their features, which is the similarity scores of modules. Scores of TM, Name and SMT modules are used here.

$$(d_i, d_j) \mapsto \left(BiTM(d_i, d_j), Names(d_i, d_j), \mathrm{SMT}(d_i, d_j) \right) \quad (7)$$

Finally, we use a simple linear combination of these features as the final score for the document pairs:

$$Score(d_i,d_j) := W_{TM} \times BiTM(d_i,d_j) + W_{Name} \times Names(d_i,d_j) + W_{SMT} \times SMT(d_i,d_j) \quad (8)$$

In this equation the scores for each pair of documents (d_i,d_j) is used: $BiTM(d_i,d_j)$ is the BiTM score, $Names(d_i,d_j)$ is the Name score, $\mathrm{SMT}(d_i,d_j)$ is two score BLEU and PER of SMT module. The weight of each model is tuned on a development set using Least Square Error approach.

4 Experiments

4.1 Training data

The available data set is a very large corpus of comparable documents, coming from the BUCC shared task. The documents are Wikipedia pages without any links, special tags or Metadata.
Training data (train.en/de) is a corpus of linked comparable documents with about 147(K) documents. The non-linked data are a set of about 166(K) English documents that have no similar document in German document space. Test set (test.en/de) is a random subset of training data that we use its *de* side as the source while ignoring the *en* side. Also, the tuning set for system combination parameters is about 1(K) documents of the training data that are not seen in the test set. Statistical information of data is reported in Table 1.

	#Documents	#Running Words	Lexicon Size
total.en	313471	264(M)	2(M)
train.en	147474	83(M)	1(M)
train.de	147474	121(M)	1(M)
test.en	10000	8(M)	239(K)
test.de	10000	5(M)	263(K)

Table 1. Statistical information of data.

4.2 Preprocessing

The first step of our work is preprocessing the input documents. So that for tokenization and normalization we use the E4SMT tools) Jabbari et al., 2012). This tool normalizes different character representations to be uniform, tokenizes the input text and also tags the specific tokens like numbers, dates, abbreviations, URL addresses, etc. In addition, the compound words of *de* side needed to be split. We have used Cdec tools for this purpose(Dyer, 2009).

4.3 Preparing modules

In this section, we introduce the tools and corpora used for training and preparing each of the modules.

4.3.1 Doc2Vec

Training Doc2Vec module consists of two steps. First we need to train a words vector model. Since the quality of word2vec model depends on the size of the training data, we train our model on all documents in the training and test sets. After this step, we train paragraph vector model and convert each document in source and target test sets to a 200-dimensional vector. After that by selecting 5000 random aligned documents from training set we calculated our transformation matrix by minimizing the error rate on those documents. Training and querying this model for all German documents can be done in several hours. The precision of this module is not very high. Hence, it cannot be used in an effective manner for predicting documents alignment. But due to its speed it can be a great filter for our other modules. The results of this experiment are shown in Figure 4.

4.3.2 BiTM

For training bilingual topic models, we use Mallet toolkit (McCallum, 2002). One important decision in topic modeling is finding the number of topics and the hyper-parameters, because of their significant impact on the resulting topic assignments. For finding the number of topics, we calculate perplexity, which is a way of evaluating the predictive power of the model (Figure 3). From now on, in all the experiments of BiTM we set number of topics to 1200.

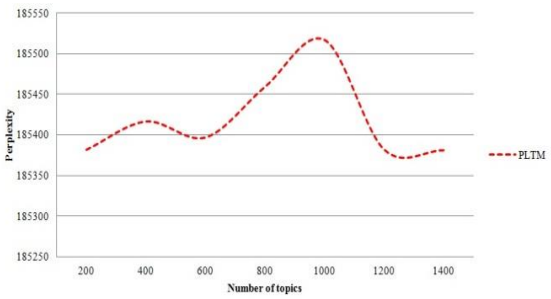
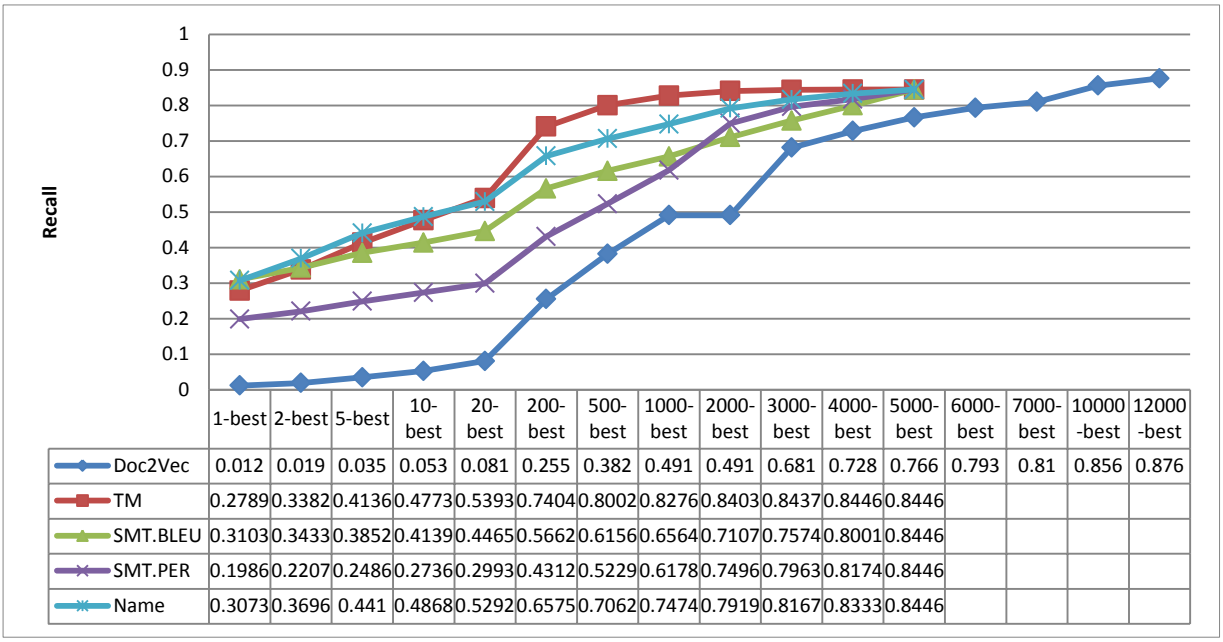


Figure 3. Perplexity for different number of topics. when $\boldsymbol{\alpha = 1}$ and $\boldsymbol{\beta = 0.7}$, the lower perplexity is better.

Also, we use the method in (Wallach, 2009) to find hyper-parameters.



	1-best	2-best	5-best	10-best	20-best	200-best	500-best	1000-best	2000-best	3000-best	4000-best	5000-best	6000-best	7000-best	10000-best	12000-best
Doc2Vec	0.012	0.019	0.035	0.053	0.081	0.255	0.382	0.491	0.491	0.681	0.728	0.766	0.793	0.81	0.856	0.876
TM	0.2789	0.3382	0.4136	0.4773	0.5393	0.7404	0.8002	0.8276	0.8403	0.8437	0.8446	0.8446				
SMT.BLEU	0.3103	0.3433	0.3852	0.4139	0.4465	0.5662	0.6156	0.6564	0.7107	0.7574	0.8001	0.8446				
SMT.PER	0.1986	0.2207	0.2486	0.2736	0.2993	0.4312	0.5229	0.6178	0.7496	0.7963	0.8174	0.8446				
Name	0.3073	0.3696	0.441	0.4868	0.5292	0.6575	0.7062	0.7474	0.7919	0.8167	0.8333	0.8446				

Figure 4. Diagram of modules recall in different neighborhood sizes.

4.3.3 Names

In this work, we use a German and English NER model to tag NEs. For this purpose, we use the Stanford NER tagger tool and also we use an unsupervised transliteration mining with the Moses toolkit[1] (Koehn et al., 2007).

4.3.4 SMT

For this module, we train a German to English SMT system. For this purpose, we use the Moses toolkit for training translation models and decoding, as well as SRILM[2] (Stolcke, 2002) to build the language models. Also, we used the German-English part of the Europarl[3] (Koehn, 2005) parallel corpora as the SMT's training corpora.

4.4 Evaluation

In this phase, we align the documents of test.de set with a proper *en* document from the collection of English documents. In the two filtering steps of the model pipeline, we reduce the size of the target space from 313(K) documents to 5(K) documents for each *de* document. The first filter is the Doc2Vec module, which is the fastest module in our model. This filter reduces the target space to 12(K) English documents that are the nearest documents to the *de* one with 87.6% of accuracy. The second filter is the Name module. This filter reduces the size of the target space from 12(K) documents to 5(K) documents with the accuracy of 84.46%.

Each *de* document in the test set is evaluated with the filtered *en* documents (5K documents). Then the vector of the similarity scores for each pair is produced and the score of the system combination module is computed for each pair of documents. The result is a matrix of similarity values. Finally, for each row of this matrix the 5-best results are extracted.

The precision, recall and F-measure for the 1-best output of the system combination module and the 5-best results list are shown in Table 2.

	5-best results	1-best System Combination
Precision	12.6	45.67
Recall	62.98	45.67
F-measure	21	45.67

Table 2. Results of our model; Precision, Recall and F-measure for 1-best System Combination and 5-best results list.

The final precision of our model is about 12%, this is because of the variation of the modules votes. Each module considers the *en* documents from a different view so the 5-best list of the final results contains the most similar *en* documents to the de input. But from this list just one of them is the exact translation. Although each Wikipedia page has a specific equivalent page in the target language but, it is probable that a set of pages are highly similar to it, especially for pages with related topics. So, because of this characteristic of Wikipedia pages, deciding the exact

[1] https://github.com/moses-smt/mosesdecoder
[2] http://www.speech.sri.com/projects/srilm/
[3] http://www.statmt.org/europarl/

translation just with the content information is a vague task. Also aligning the *de* document to a proper *en* one from a large collection of *en* samples increases this ambiguity.

5 Conclusion

Our work is a framework consists of several modules for retrieving similar Wikipedia pages for German documents from a large collection of English documents. Our model is proper for dealing with very large corpora. The results show that our model is able to find the correct answer in 62% of samples.

The framework proposed here has two advantages over the previous works: firstly it can handle searching through a large collection of data which is achieved by applying the filtering modules. And also everything was done just by using the content information of documents, without using any special tags or Metadata.

Also, all of the modules used in our framework are language independent, and it could be used for any other language pairs.

Acknowledgement

This research was partially supported by targoman.com. We thank our colleagues from targoman.com, who provided insight and expertise that greatly assisted the research.

References

Barrón-Cedeno, Alberto, Paolo Rosso, David Pinto, and Alfons Juan. (2008). On Cross-lingual Plagiarism Analysis using a Statistical Model. In PAN.

Blei, David M., Andrew Y. Ng, and Michael I. Jordan. (2003). Latent dirichlet allocation. the Journal of machine Learning research, 3, 993-1022.

Brown, Peter F., Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. (1993) .The mathematics of statistical machine translation: Parameter estimation. Computational linguistics 19, no. 2, 263-311.

Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze. (2008). An Introduction to information retrieval. Vol. 1. Cambridge: Cambridge university press.

Deerwester, Scott C., Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. (1990). Indexing by latent semantic analysis. JAsIs 41, no. 6, 391-407.

Delpech, Estelle. (2011). Evaluation of terminologies acquired from comparable corpora: an application perspective. In Proceedings of the 18th International Nordic Conference of Computational Linguistics (NODALIDA 2011), pages 66-73.

Dumais, Susan T., Todd A. Letsche, Michael L. Littman, and Thomas K. Landauer. (1997, March). Automatic cross-language retrieval using latent semantic indexing. In AAAI spring symposium on cross-language text and speech retrieval (Vol. 15, p. 21).

Durrani, Nadir, Hieu Hoang, Philipp Koehn, and Hassan Sajjad. (2014). Integrating an unsupervised transliteration model into statistical machine translation. EACL 2014, 148.

Dyer, Chris. (2009). Using a maximum entropy model to build segmentation lattices for MT. In Proceedings of NAACL HLT 2009, Boulder, Colorado.

Franco-Salvador, Marc, Paolo Rosso, and Roberto Navigli. (2014, April). A knowledge-based representation for cross-language document retrieval and categorization. InProceedings of EACL (pp. 414-423).

Fung, Pascale, and Percy Cheung. (2004). Multi-level bootstrapping for extracting parallel sentences from a quasi-comparable corpus. In Proceedings of the 20th international conference on Computational Linguistics, page 1051. Association for Computational Linguistics.

Gupta, Rajdeep, and Sivaji Bandyopadhyay. (2013). Testing the Effectiveness of Named Entities in Aligning Comparable English-Bengali Document Pair. In Intelligent Interactive Technologies and Multimedia (pp. 102-110). Springer Berlin Heidelberg.

Jabbari, Fattaneh, Somayeh Bakhshaei, Seyed Mohammad Mohammadzadeh Ziabary, and Shahram Khadivi. (2012). Developing an Open-domain English-Farsi Translation System Using AFEC: Amirkabir Bilingual Farsi-English Corpus. In The Fourth Workshop on Computational Approaches to Arabic Script-based Languages (p. 17).

Kilgarriff, Adam. (2001). Comparing corpora. International journal of corpus linguistics, 6, no. 1 (pp. 97-133).

Knoth, Petr, Lukas Zilka, and Zdenek Zdrahal. (2011). Using explicit semantic analysis for cross-lingual link discovery. In Proceedings of the 5th International Joint Conference on Natural Language Processing , pages 2-10.

Koehn, Philipp, Franz Josef Och, and Daniel Marcu. (2003). Statistical phrase-based translation. In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1, pp. 48-54.

Koehn, Philipp. (2005). Europarl: A parallel corpus for statistical machine translation. MT summit, (pp. 79-86).

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar,

Alexandra Constantin, Evan Herbst. (2007). Moses: Open source toolkit for statistical machine translation. Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions, (pp. 177-180).

Le, Quoc V., and Tomas Mikolov. (2014). Distributed Representations of Sentences and Documents. Int. Conf. Mach. Learn. ICML 2014, vol. 32, pp. 1188–1196.

Li, Bo, and Eric Gaussier. (2010). Improving corpus comparability for bilingual lexicon extraction from comparable corpora. In Proceedings of the 23rd International Conference on Computational Linguistics, pages 644-652. Association for Computational Linguistics.

McCallum, Andrew K.. (2002). MALLET: A Machine Learning for Language Toolkit. http://mallet.cs.umass.edu.

Mcnamee, Paul, and James Mayfield. (2004). Character n-gram tokenization for European language text retrieval. Information retrieval 7, no. 1-2 (pp. 73-97).

Mehdizadeh Seraj, Ramtin, Fattaneh Jabbari, and Shahram Khadivi. (2014). A novel unsupervised method for named-entity identification in resource-poor languages using bilingual corpus. In Telecommunications (IST), 2014 7th International Symposium on (pp. 519-523). IEEE.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. (2013). Distributed representations of words and phrases and their compositionality. In Advances in Neural Information Processing Systems, pp. 3111-3119.

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

Mikolov, Tomas, Quoc V. Le, and Ilya Sutskever. (2013). Exploiting similarities among languages for machine translation." arXiv preprint arXiv:1309.4168.

Mimno, David, Hanna M. Wallach, Jason Naradowsky, David A. Smith, and Andrew McCallum. (2009, August). Polylingual topic models. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2 (pp. 880-889). Association for Computational Linguistics.

Mousavi Nejad, Najmeh, and Shahram Khadivi. (2011). An Unsupervised Alignment Model for Sequence Labeling: Application to Name Transliteration. 2011 Named Entities Workshop.

Navigli, Roberto, and Simone Paolo Ponzetto. (2012). BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network.Artificial Intelligence, 193, 217-250.

Nießen, Sonja, Franz Josef Och, Gregor Leusch, and Hermann Ney. (2000). An Evaluation Tool for Machine Translation: Fast Evaluation for MT Research. Proceedings of the 2nd International Conference on Language Resources and Evaluation. Athens, Greece.

Oard, Douglas W. (1998). A comparative study of query and document translation for cross-language information retrieval. Machine Translation and the Information Soup. Springer Berlin Heidelberg, 472-483.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. (2002). Bleu: a Method for Automatic Evaluation of Machine Translation. In Proceedings of the 40th annual meeting on association for computational linguistics (pp. 311-318). Association for Computational Linguistics.

Potthast, Martin, Benno Stein, and Maik Anderka. (2008). A Wikipedia-based multilingual retrieval model. In Advances in Information Retrieval (pp. 522-530). Springer Berlin Heidelberg.

Smith, Jason R., Chris Quirk, and Kristina Toutanova. (2010). Extracting parallel sentences from comparable corpora using document level alignment. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pages 403-411. Association for Computational Linguistics.

Snover, Matthew, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. (2006). A study of translation edit rate with targeted human annotation. Proceedings of association for machine translation in the Americas.

Steyvers, Mark, and Tom Griffiths. (2007). Probabilistic topic models. Handbook of latent semantic analysis 427(7), 424-440.

Stolcke, Andreas. (2002). SRILM-an extensible language modeling toolkit. INTERSPEECH.

Su, Fangzhong, and Bogdan Babych. (2012). Development and Application of a Cross-language Document Comparability Metric. In *LREC*, (pp. 3956-3962).

Tang, Guoyu, Yunqing Xia, Min Zhang, Haizhou Li, and Fang Zheng. (2011). CLGVSM: Adapting Generalized Vector Space Model to Cross-lingual Document Clustering. In IJCNLP , pages 580-588.

Tillmann, Christoph, Stephan Vogel, Hermann Ney, Arkaitz Zubiaga, and Hassan Sawaf. (1997). Accelerated DP based search for statistical translation. Eurospeech.

Wallach, Hanna M., David Minmo, and Andrew McCallum. (2009). Rethinking LDA: Why priors matter.

Winograd, Terry. (1972). Understanding natural language. *Cognitive psychology*, *3*(1), 1-191.

Zens, Richard, Franz Josef Och, and Hermann Ney. (2002). Phrase-based statistical machine translation. In KI 2002, Advances in Artificial Intelligence (pp. 18-32). Springer Berlin Heidelberg.