Cross-Language Retrieval Using Link-Based Language Models

Benjamin Roth
beroth@coli.uni-saarland.de

Dietrich Klakow
dietrich.klakow@lsv.uni-saarland.de

Spoken Language Systems, Saarland University
D-66125 Saarbrücken, Germany

ABSTRACT

We propose a cross-language retrieval model that is solely based on Wikipedia as a training corpus. The main contributions of our work are: 1. A translation model based on linked text in Wikipedia and a term weighting method associated with it. 2. A combination scheme to interpolate the link translation model with retrieval based on Latent Dirichlet Allocation. On the CLEF 2000 data we achieve improvement with respect to the best German-English system at the bilingual track (non-significant) and improvement against a baseline based on machine translation (significant).

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*retrieval models*

General Terms

Algorithms, Experimentation

Keywords

CLIR, Wikipedia, LDA, language modeling

1. INTRODUCTION

Translation lexica and parallel corpora are often only accessible for some European language pairs. And even if they exist their vocabulary is inherently limited in contrast to an ever growing wide-coverage ressource like Wikipedia, where corresponding articles are connected across languages. Attempts have therefore be made to extract information for CLIR from article-level co-occurrence statistcs, with moderate succes [6]: Coarse thematical relationships alone can arguably not capture the specific meaning contained in a query, a word-to-word translation is necessary. The question therefore arises how word-specific mappings can be obtained from freely available large-scale knowledge sources such as Wikipedia.

Several approaches in this direction have been undertaken. Often, a Wikipedia title in the source language is associated with a word and the corresponding title in the target language is used as a translation [4]. However, [5] note that the vocabulary distribution of titles has a skew to certain words. Because the titles of Wikipedia articles are specifically tailored to be unique identifiers rather than representative text samples, translations of large classes of words might be problematic when a variation is used. In [2] a bilingual dictionary is extracted from Wikipedia by supervised classification.

The method we use is unsupervised and based on the anchor text of links. In Wikipedia, any text can be linked to any page. For example, the German texts "in Wäldern gelegte Brände" and "Buschbrand" may be valid contexts to be linked to the article with the English counterpart "Wildfire". Three items of information are necessary to build a probabilistic translation model based on linked text:

- How likely is a text to be linked?
- What is a probable target article, given a linked text?
- What is a probable anchor text, given a link to a certain article?

2. LINK TRANSLATION MODEL

If, for the sake of simplicity, a unigram model is used for translation this amounts to the probablities $P(l|w)$, $P(a|w,l)$ and $P(w|a,l)$. In a bilingual setting l is a variable indicating whether the source word is linked, a is the bilingual article the source word is linked to, and w_E and w_F are words in the source and the target languages respectively. The probability of translating a source word into a target word is:

$$\begin{aligned} P(w_F|w_E) &= P(w_F|l_{\text{true}}, w_E)P(l_{\text{true}}|w_E) \\ &\quad + P(w_F|l_{\text{false}}, w_E)P(l_{\text{false}}|w_E) \end{aligned}$$

We focus on the linked case. Assuming that the translation of linked source words does only depend on the articles they are linked to, one gets:

$$P(w_F|l_{\text{true}}, w_E) = \sum_a P(w_F|a, l_{\text{true}})P(a|l_{\text{true}}, w_E)$$

We note that the probability $P(l_{\text{true}}|w_E)$ can function as a term weighting in the source documents, assuming that the importance of terms is correlated with their probability of being linked. Translation and linking are assumed to be independent of the document, given a source word. In the following we write D for a source document in language E, l for l_{true} and $n(w,Q)$ for the count of w in a query Q.

2.1 The Pure Link Model

In a query likelihood model a document provides a probabilistic model for a query. The ranking is usually done by $\log P(Q|D) = \sum_{w \in Q} n(w,Q) \log P(w|D)$.

SIGIR'10, July 19–23, 2010, Geneva, Switzerland.
ACM 978-1-60558-896-4/10/07.

In principle, all components are provided by the link model to perform retrieval in such a setup. The probability that Wikipedia article a is the link target when a linked word is picked from document D (making the same independence assumptions as before) is

$$P(a|D,l) = \frac{\sum_{w_E} P(a|l,w_E)P(l|w_E)P(w_E|D)}{\sum_{a',w_E} P(a'|l,w_E)P(l|w_E)P(w_E|D)}$$

and $P(w_F|a,l)$ is the probability that, given a source word is linked to article a, it is translated to word w_F. Together the elements of the link model provide us with the distribution:

$$P_{link}(w_F|D,l) = \sum_a P(w_F|a,l)P(a|D,l)$$

It is practical to think of this distribution as combined in that way, because it separates the per-document estimates from the vocabulary estimates. The atomic probabilities are estimated from relative frequencies: $P(a|l,w_E)$, $P(l|w_E)$ and $P(w_F|a,l)$ from Wikipedia, $P(w_E|D)$ from the current document.

This formulation poses three problems: **The zero-probability problem:** Because the components of the model are estimated from relative frequencies, to many events zero probability is assigned. **The summation problem:** If the probability distributions are smoothed and are never zero, summation might for every word go over all Wikipedia articles, which would be prohibitively expensive to compute. **The training basis problem:** The model only considers words likely to be linked. Especially high frequency or function words could have skewed distributions.

The model is hence not immediately applicable. We tackle these problems by interpolating the link model with a language model based on LDA and by considering the probability of being linked for the query term weights.

2.2 Model Combination on Word Level

One possible combination scheme of LDA and link model is to interpolate word distributions given a document. We use Wikipedia as a bilingual training corpus for LDA by cutting articles at 100 words, discarding shorter ones and concatenating both language sides. We trained models with 125, 250, 500 and 1000 topics (parameterized as suggested in [3]) and interpolated them with equal weight to avoid local maxima. After inference on the retrieval collection, one has:

$$P_{LDA}(w|D) = \sum_z P(w|z)P(z|D)$$

Having the word probabilities, the question arises how to weight the query word counts, as a weighting according to their probability of being linked seems reasonable for the link model, but is not justified for LDA. Hence, we use two model parameters α and β to interpolate weightings and distributions respectively. The query log-likelihood becomes:

$$\begin{aligned}\log P_{\alpha,\beta}(Q|D) &= \sum_{w\in Q} n(w,Q)\left[\alpha P(l|w) + (1-\alpha)\right] \\ &\quad \cdot \log\left[\beta P_{link}(w|D,l) + (1-\beta)P_{LDA}(w|D)\right]\end{aligned}$$

The LDA-distributions are smooth by defininition, so for $0 \leq \beta < 1$ there is no zero-probability problem. For efficiency reasons, we did not smooth the link component and summed only over the 1000 most probable articles per document. In an ad-hoc parametrization we let both weightings and both models contribute equally strongly. The probability of a word being linked is very low with $p(l) = .06$, to get equal influence of both weightings we require $\alpha \cdot p(l) = 1-\alpha$, which results in $\alpha = .94$, β is set to .5.

We evaluated on the German-English CLEF 2000 bilingual track[1] (title+description) and achieve *map*= .291 which is better than any of those reported for the same language pair [1], this difference is however statistically not significant. The model is significantly ($p < 0.05$, paired t-test) better than a base-line using Moses machine translation trained on Europarl with tf.idf vector retrieval.

Table 1: Results on German-English CLEF2000

method	map	gmap
LDA only	.074	.002
Moses + tf.idf	.203	.061
Best system CLEF2000	.267	–
LDA + links	**.291**	**.107**

3. CONCLUSION

We have introduced a CLIR method that is based solely on information extracted from Wikipedia. It combines document-level information (captured by LDA) and word-specific information (captured by a link model) in a clear language modeling setup. There is much room for the exploration of different smoothing and combination schemes. We tried a simple one on word level. This model did not only outperform a base-line obtained with Moses machine translation, it also produced results that compare favorably against values reported for the CLEF 2000 bilingual track.

4. REFERENCES

[1] M. Braschler. CLEF 2000-Overview of results. In *Cross-language information retrieval and evaluation: workshop of the Cross-Language Evaluation Forum*. Springer Verlag, 2001.

[2] M. Erdmann, K. Nakayama, T. Hara, and S. Nishio. Improving the extraction of bilingual terminology from Wikipedia. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2009.

[3] T. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(90001):5228–5235, 2004.

[4] D. Nguyen, A. Overwijk, C. Hauff, R. Trieschnigg, D. Hiemstra, and F. de Jong. WikiTranslate: Query translation for cross-lingual information retrieval using only Wikipedia. In *9th Workshop of the Cross-Language Evaluation Forum*, 2008.

[5] J. Sjobergh, O. Sjobergh, and K. Araki. What types of translations hide in wikipedia? *Lecture Notes in Computer Science*, 4938:59, 2008.

[6] P. Sorg and P. Cimiano. Cross-lingual information retrieval with explicit semantic analysis. In *Working Notes of the Annual CLEF Meeting*, 2008.

[1]ELRA catalogue (`http://catalog.elra.info`), The CLEF Test Suite for the CLEF 2000-2003 Campaigns, catalogue reference: ELRA-E0008; All data was processed using the Snowball stemmer and stopword list.