Czech Information Retrieval with Syntax-based Language Models

Jana Straková, Pavel Pecina

Institute of Formal and Applied Linguistics
Charles University in Prague
strakova@ufal.mff.cuni.cz, pecina@ufal.mff.cuni.cz

Abstract

In this paper, we deal with information retrieval approach based on language model paradigm, which has been intensively investigated in recent years. We propose, implement, and evaluate an enrichment of language model employing syntactic dependency information acquired automatically from both documents and queries. By testing our model on the Czech test collection from Cross Language Evaluation Forum 2007 Ad-Hoc track, we show positive contribution of using dependency syntax in this context.

1. Introduction

In recent years, considerable attention has been dedicated to language modeling methods in information retrieval (Ponte and Croft, 1998). Although these approaches generally allow exploitation of any type of language model, most of the published experiments were conducted with a classical n-gram model, usually limited only to unigrams. A few works exploiting syntax in information retrieval can be cited in this context (Lee and Lee, 2005), (Nallapati and Allan, 2002), (Gao et al., 2004), but significant contribution of syntax based language modeling for information retrieval is yet to be proved.

Our experiments are conducted on Czech which is a morphologically rich language and has a considerably free word order. Especially, the long distance relations between words are expected to be captured better in a syntactic language model than in a bigram language model based on surface word order.

The paper is organized as follows. First, we describe the test collection (Section 2) and the methodology (Section 3) used in our work. Experiments and their results are described in Section 4 and discussed in Section 5. The work is concluded in Section 6.

2. Test collection

For our experiments, we used Czech test collection from Cross Language Evaluation Forum 2007 Ad-Hoc Track (CLEF, 2007) consisting of $81,735$ documents (news articles) and relevance assessments of 50 topics. The average length of documents is 349.76 words and 15.24 documents in average are assessed as relevant to each topic. The topics are presented in TREC format as a structure of three fields describing each topic by a keyword query (`title`) and in more detail by a few sentences (`narr` and `desc`).

For development and evaluation purposes, we randomly divided these 50 topics into a development set of 10 topics and test set of 40 topics.

This test collection was used at the CLEF 2007 Ad-Hoc track and some evaluation results using this collection have already been published. An overview of the CLEF 2007 Ad-Hoc results can be found in (Nunzio et al., 2008).

3. Methodology

3.1. Notation

Throughout the paper, we will be using the following notation: D stands for a document and C for a collection of documents, which we rank by relevance to a query Q. A query Q consists of terms $Q = q_1, q_2, \ldots, q_n$, thus a bigram of two subsequent terms (by subsequent we mean "subsequent on surface") is (q_i, q_{i+1}). A dependency relation between two words is denoted as $(p(q_i), q_i)$, where $p(q_i)$ is the head word and q_i its modifier.[1] For an example of a dependency tree, see figure 1.

3.2. Language modeling in information retrieval

In language modeling based information retrieval, for each query Q, all documents D from the collection C are ranked by the probability $P(D|Q)$ of being (independently) generated by the query language model. From the Bayes formula and the fact that $P(Q)$ is constant for all documents and $P(D)$ is considered uniform across the collection, we can rank the documents by the "reverted" probability $P(Q|D)$ with the same result. Thus, instead of estimating probability of document D being generated by language model defined by Q, we will consider the probability of query Q being generated from the language model defined by document D.

Having introduced our key ranking function, $P(Q|D)$, we will simplify the notation $P(D|Q)$ to $P_D(Q)$. Furthermore, since $Q = q_1, \ldots, q_i, \ldots, q_n$, the probability of a single term q_i is denoted as $P_D(q_i)$. Similarly, $P_D(q_i, q_{i+1})$ and $P_D(p(q_i), q_i)$ stand for the probability of a surface bigram and dependency bigram (respectively) in a language model of document D.[2]

In the following formulas, P_D stands for document probability, such as $P_D(Q)$ is the probability of the whole query

[1] We assume the pair (q_i, q_{i+1}) to be ordered, therefore the word first appearing in the sentence q_i takes the first position in the ordered pair. Similarly, in the dependency bigram $(p(q_i), q_i)$ the first position in the pair is taken by the head $p(q_i)$ and the second one by the modifier q_i.

[2] Here we should write properly $P_D((q_i, q_{i+1}))$ and $P_D((p(q_i), q_i))$, but we took the liberty of removing the extra pair of brackets as we believe there is no danger of confusion.

Q given document D and $P_D(q_i)$ is the probability of single term q_i in language model defined by document D. C_D stands for raw counts (frequencies) of the corresponding language phenomena, e.g. $C_D(q_i)$ is the frequency of term q_i in document D. $|D|$ denotes the size of the document with respect to the current model, therefore in unigram model, it is a number of unigrams, in surface bigram model, it is number of surface bigrams and finally in dependency bigram model, it denotes the number of dependency relations defined by dependency syntax tree representing the document (sentences), as will be more precisely explained in the next chapter 3.3..
Apart from the well known unigram model

$$P_D^{unigram}(Q) = \prod_{i=1}^{n} P_D(q_i) \hat{=} \prod_{i=1}^{n} \frac{C_D(q_i)}{|D|}$$

and surface bigram model

$$P_D^{surf.bigram}(Q) = \prod_{i=1}^{n-1} P_D(q_i, q_{i+1}) \hat{=} \prod_{i=1}^{n-1} \frac{C_D(q_i, q_{i+1})}{|D|}$$

we will also use a dependency bigram model described in the following subsection.

3.3. Dependency bigram model

In dependency syntax (as it is used in this work), the sentence structure is represented as a tree with nodes formed by words and edges determined by relations between words. Thus, there is a bijection between words of the sentence and nodes of the tree and each word has one parent, except for the root, which is usually the predicate of the sentence (the verb). An example of a dependency tree is shown below on Figure 1.

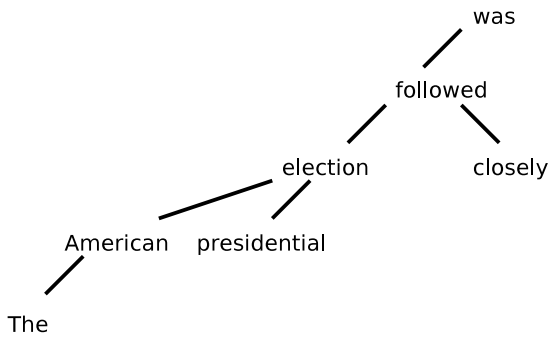

Figure 1: An example of a dependency tree for sentence "The American presidential election was followed closely."

For each sentence in query $Q = q_1, \ldots, q_n$, we build its dependency tree so each word in the query, except for sentence roots, has a parent and define:

$$P_D^{dep.bigram}(Q) = \prod_{q_i : \exists p(q_i)} P_D(p(q_i), q_i)$$

which using MLE can be estimated as

$$P_D^{dep.bigram}(Q) \hat{=} \prod_{q_i : \exists p(q_i)} \frac{C_D(p(q_i), q_i)}{|D|}$$

In this case, $|D|$ stands for "number of syntactic (dependency) relations" in the document, as opposed to "number of unigrams, bigrams" in unigram or bigram model, respectively.
We smoothed all document probabilities by linear interpolation with collection probabilities, which is known as Jelinek-Mercer smoothing (Jelinek and Mercer, 1980):

$$\tilde{P}_D(q_i) = \lambda P_D(q_i) + (1 - \lambda) P_C(q_i), \lambda \in \langle 0, 1 \rangle$$

In this formula, $P_D(q_i)$ is the probability of term q_i in language model defined by document D, and similarly, $P_C(q_i)$ is the probability of term q_i in the collection C of all documents.

4. Experimental setup

As a baseline, we used the plain unigram model. The proposed dependency bigram model is also compared with the classical surface bigram model to prove the hypothesis that more information is captured by dependency bigrams than by surface bigrams.
In many information retrieval systems, especially when dealing with morphologically rich languages, some form of stemming is used. In our experiments, we employ lemmatization as a linguistically motivated means of stemming. Lemmatization is a process of mapping a word to its base form (lemma), such as infinitives for verbs. Of course, we could replace the lemmatization step with stemming if we so prefer. In our case, lemmatization was a product of the preprocessing tagging step for the dependency parsing.
Thus, by combination of unigram, surface bigram and dependency bigram model and their lemmatized and non-lemmatized versions, there are six models to evaluate.
Finally, we combined these six models by means of a simple linear interpolation. The linear coefficients were estimated by grid search with MAP as objective function on a development set of 10 topics. The optimal coefficients for each of the six models are shown in table 1.
Morphological analysis (including tagging and lemmatization) was performed with Feature-based tagger (Hajič, 2004) and dependency syntax parsing with MST Parser (McDonald et al., 2005) in TectoMT framework (Žabokrtský et al., 2008).
Since we are depending on syntactic information in our work, we used all three sections of (`title`, `narr` and `desc`) of the TREC-style description of the topics (queries) to be able to benefit from the linguistic information underlying a longer, natural language text.
As stopwords set, we used 256 Czech stopwords available at (UniNE, 2005).
We also performed pseudo relevance feedback on highly ranked documents as a method of broadening the query with semantically relevant terms (Manning et al., 2008).
As evaluation measure, we use common Mean Average Precision (MAP) computed by evaluation tool (trec_ eval, 2008).

5. Results and Discussion

Figure 2 and Table 1 show results of the six models: unigram model with non lemmatized word forms, unigram model with lemmas, surface bigram model with non

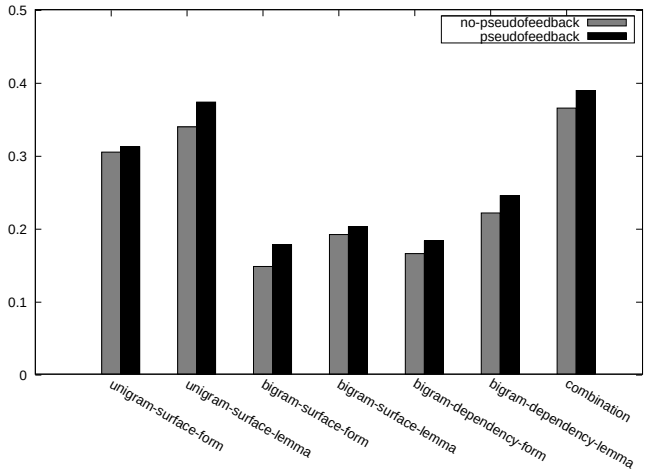

Figure 2: MAP for language models and their combination.

lemmatized word forms, surface bigram model with lemmas, dependency bigram model with non lemmatized word forms and dependency bigram model with lemmas. The last model is created by a linear interpolation of all models where the coefficients have been estimated by simple grid search on a development set of 10 topics.

The results presented in figure 2 and table 1 have been evaluated on test set of 40 topics.

As for single models, the unigram model reaches the highest values of MAP. This result is expected, as the higher order n-gram models alone are often too specific for the given task. However, in combination with the unigram model, we observe increased performance of the system. The combination of all language models reaches MAP 0.3890. For comparison purposes, the MAP computed on all 50 topics is 0.4102. This result outperforms most of the models presented in (Nunzio et al., 2008) and is close to the best published result on this test collection, using solely means of language modeling. However, it must be noted that the comparison is somewhat problematic due to the optimization performed on a subset of 10 topics.

Intriguingly, the dependency bigram model outperforms the bigram surface model. The explanation for this is presented in figure 3 which shows MAP for particular topics evaluated in the system for both models. An interesting observation about figure 3 is that rather than improving the results for each topic constantly, the dependency model performs noticeably better on certain topics. To elaborate on that, dependency model seems to pick bigrams with higher information content than the surface bigram model.

Let us pick an example, where the difference is particularly remarkable: a topic 7 "Australský premiér (Australian prime minister)". By inspecting the bigrams participating highly in ranking of the documents, we find surface bigrams "být, v (be, in)", "být kdo (be, who)", "australský premiér (australian prime minister)", "být který (be, who)", whereas the dependency bigram model employs bigrams "australský primér (australian prime minister)" and "být kdo (be, who)". Please note that here we are working with lemmatized version of the model, hence the lemma "be". We assume that the dependency bigram model is more successful in implicitly selecting the correct bigrams and weighting them properly. Apparently, the bigram surface model bigrams can be pruned by good stopword list, but given the flective nature of the Czech language, there were always useless word forms, which even a broadened stoplist did not manage to prune.

This might be caused by the very definition of the dependency syntax tree and the fact that bigrams in dependency model are formed of head-modifier word pairs rather than of word pairs brought together by sentence word order.

6. Conclusions

We have presented a simple dependency bigram language model as an extension of commonly used unigram and bigram surface model. With this language model, we have outperformed most of the results published in (Nunzio et al., 2008). Finally, we have found examples, where the dependency bigram model produced significantly better output than surface bigram model.

7. References

CLEF. 2007. Cross Language Evaluation Forum (CLEF), http://clef-campaign.org.

Jianfeng Gao, Jian-Yun Nie, Guangyuan Wu, and Guihong Cao. 2004. Dependence language model for information retrieval. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 170–177, New York, NY, USA. ACM.

Jan Hajič. 2004. *Disambiguation of Rich Inflection (Computational Morphology of Czech)*, volume 1. Charles University Press, Prague.

Frederick Jelinek and Robert L. Mercer. 1980. Interpolated estimation of markov source parameters from sparse data. In *Proceedings of the Workshop on Pattern Recognition in Practice*, Amsterdam, The Netherlands: North-Holland, May.

Changki Lee and Gary Geunbae Lee. 2005. Probabilistic information retrieval model for a dependency structured indexing system. *Inf. Process. Manage.*, 41(2):161–175.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schtze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Human Language Technologies and Empirical Methods in Natural Language Processing (HLT-EMNLP)*, Vancouver, Canada.

Ramesh Nallapati and James Allan. 2002. Capturing term dependencies using a language model based on sentence trees. In *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*, pages 383–390, New York, NY, USA. ACM.

Giorgio M. Nunzio, Nicola Ferro, Thomas Mandl, and Carol Peters. 2008. Clef 2007: Ad hoc track overview. pages 13–32.

Jay M. Ponte and W. Bruce Croft. 1998. A language modeling approach to information retrieval. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR*

model	no feedback	feedback	coefficient
unigram-surface-form	$0,3046$	$0,3116$	0.1
unigram-surface-lemma	$0,3392$	$0,3731$	0.65
bigram-surface-form	$0,1477$	$0,1775$	0.05
bigram-surface-lemma	$0,1915$	$0,2023$	0.05
bigram-dependency-form	$0,1654$	$0,1826$	0.05
bigram-dependency-lemma	$0,2211$	$0,2447$	0.1
combination	$0,3650$	$\mathbf{0,3890}$	

Table 1: MAP for language models and their combination

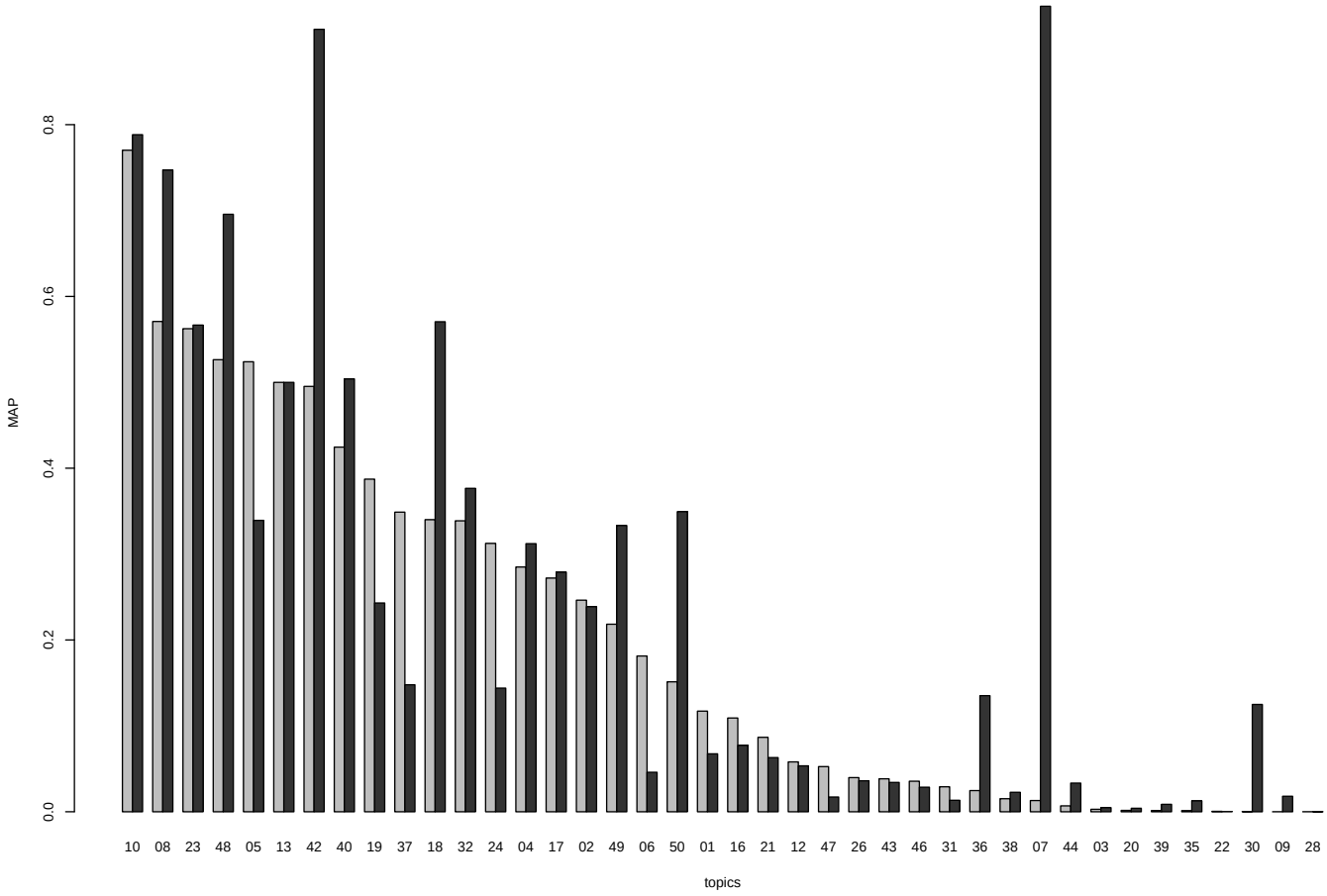

Figure 3: Comparison of surface (gray) and dependency (black) lemmatized bigram models on particular topics.

conference on Research and development in information retrieval, pages 275–281, New York, NY, USA. ACM.

trec_eval. 2008. http://trec.nist.gov/trec_eval.

UniNE. 2005. IR multilingual resources at UniNE, http://www.unine.ch/info/clef/.

Zdeněk Žabokrtský, Jan Ptáček, and Petr Pajas. 2008. TectoMT: Highly Modular MT system with tectogrammatics used as a transfer layer. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 167–170. Association for Computational Linguistics, June.