Inf Retrieval (2014) 17:21–51
DOI 10.1007/s10791-013-9220-9

Latent word context model for information retrieval

Bernard Brosseau-Villeneuve · Jian-Yun Nie · Noriko Kando

Received: 15 September 2012 / Accepted: 4 February 2013 / Published online: 5 March 2013

Abstract The application of word sense disambiguation (WSD) techniques to information retrieval (IR) has yet to provide convincing retrieval results. Major obstacles to effective WSD in IR include coverage and granularity problems of word sense inventories, sparsity of document context, and limited information provided by short queries. In this paper, to alleviate these issues, we propose the construction of latent context models for terms using latent Dirichlet allocation. We propose building one latent context per word, using a well principled representation of local context based on word features. In particular, context words are weighted using a decaying function according to their distance to the target word, which is learnt from data in an unsupervised manner. The resulting latent features are used to discriminate word contexts, so as to constrict query's semantic scope. Consistent and substantial improvements, including on difficult queries, are observed on TREC test collections, and the techniques combines well with blind relevance feedback. Compared to traditional topic modeling, WSD and positional indexing techniques, the proposed retrieval model is more effective and scales well on large-scale collections.

Keywords Retrieval models · Word context discrimination (WCD) · Word context · Topic models · Word sense disambiguation (WSD)

1 Introduction

Single terms are often ambiguous. For example, the word "application" can mean a software application, an application for a position, and so on. However, when a term is

B. Brosseau-Villeneuve (✉) · J.-Y. Nie
University of Montréal, CP. 6128 succ. Centre-ville, Montreal, QC H3C 3J7, Canada
e-mail: bbrosseau@gmail.com

J.-Y. Nie
e-mail: nie@iro.umontreal.ca

N. Kando
National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan
e-mail: kando@nii.ac.jp

placed into context, its meaning becomes clear. The use of context is the very principle of word sense disambiguation (WSD).

In information retrieval (IR), the problem of term ambiguity is important, particularly in short queries. However, in the bag-of-words approach, each term is matched separately between the query and the document, without taking context into account. As a result, the retrieved document could contain the same terms as the query, but with different meanings (e.g. a document about "office hours for passport applications" for a query on "office application").

This problem has been observed for a long time and various approaches have been proposed to deal with it. The proposed approaches could be classified into two groups: expansive and constrictive. Both add some contextual information in the query representation.

More specifically, expansive techniques introduce additional descriptive features relevant to the context of a meaning into the query in order to favor the match with the correct context. A typical approach in this group is query expansion which adds a set of related terms into the query representation, which can be those that co-occur often with the query terms in the collection (Bai et al. 2005; Berger and Lafferty 1999; Lafferty and Zhai 2001; Xu and Croft 1996), in query logs (Cui et al. 2002) and in the retrieved documents (Cao et al. 2008; Lavrenko and Croft 2001). Even though more contextual information is added into the query representation, as the added terms are used in a bag-of-words approach, document context is not directly constrained, i.e., words can match freely any terms in the document. This can easily lead to query drift. The effectiveness of the expansion strongly depends on how the added terms are obtained.

Constrictive techniques aim to tighten the scope of the query, usually by incorporating stricter constraints. For instance, we may use more precise indexing units such as bigrams and biterms (Metzler and Croft 2004; Song and Croft 1999; Srikanth and Srihari 2002), or introduce keyword proximity information via positional indexes (Lv and Zhai 2009; Zhao and Yun 2009). One can also model dependencies between terms using techniques such as undirected graphical models (Metzler and Croft 2005) and dependency trees (Gao et al. 2004). These additional relations between query terms introduced into the query representation impose a stricter matching criterion than single-word matching. However, it is important to determine the correct relations to impose, and this is often difficult in practice. Imposing a wrong dependency could harm the retrieval process rather than help it.

Another type of constrictive technique that has yet to prove its usefulness for IR is explicit WSD (Ide and Véronis 1998; Navigli 2009). While ambiguity issues have been largely investigated for IR (Brown et al. 1991; Gaustad 2001; Gonzalo et al. 1998; Krovetz and Croft 1992; Sanderson 1994; Sanderson and Van Rijsbergen 1999; Voorhees 1993), the studies often report negative results or show some impact on collections of limited size (Sanderson 2000). The usefulness of WSD in IR still lacks strong experimental evidence.

In this paper, we will present a technique which is also grounded on WSD principles. However, in IR, the most important aspect is to distinguish between different meanings of words used in documents and in queries. This is a task of word context discrimination (WCD). In order to determine whether a term in the query has the same meaning as in a document, we can test whether its contexts in the query and in the document are similar.

An important aspect is what elements can be used to represent a word's context. WSD systems typically extract various features from the context of word occurrences: terms, collocations, syntactic relations, POS-tags and so on. For IR applications, many of such features cannot be computed reliably on search queries, as they are often short and ungrammatical. In this study, we will thus rely on simpler features, namely co-occurrence

features with the *target word*—the word whose context needs to be discriminated. We identify three types of such basic features: (1) the raw form of the target word, (2) the neighboring stop words, and (3) the context content words (word stems). Of this last type of feature, we can expect that they do not all have the same degree of constraint on the target word. Intuitively, the closer a context word to the target word, the stronger its constraint on the meaning of the latter. This principle has been used in several previous studies by defining a decaying function along with the distance (Lv and Zhai 2009). However, since it is not clear what decaying function is the most suitable, we have conducted WSD experiments to determine the appropriate decaying function using unsupervised learning (Brosseau-Villeneuve et al. 2011). It turns out that the most appropriate decaying function is similar to a power law. Using the resulting decaying function, we achieved state-of-the-art accuracy in supervised WSD while using only the three types of features stated above. In this paper, we will use the same features, for a principled and effective weighting of word context for IR.

Another problem usually encountered in context modeling is the sparseness of the features (words in our case). This is an important issue in WSD, and in IR, this issue is further amplified by short queries. For instance, using a straightforward representation by a context vector of co-occurring words, a short query will be unable to match with many of the relevant document contexts, which may use different words. However, even if the context words are different, one can generally assume that they are semantically related. To deal with the context sparsity problem, we propose to match documents and queries through latent context models created with latent Dirichlet allocation (LDA) (Blei et al. 2003). The topics act as a more abstract semantic representation of contexts. Context similarity can then be computed based on the resulting latent topics.

The proposed technique has been tested on several TREC collections. In all our experiments, this approach achieved substantial and statistically significant improvements in retrieval effectiveness over the traditional bag-of-words approach and other positional indexing techniques such as Markov random field (MRF) (Metzler and Croft 2005). Our experiments also show that our technique is complementary to techniques that use blind feedback techniques such as Relevance Model (RM) (Lavrenko and Croft 2001). Combining our approach with RM yields additive improvements. This series of experiments show that our approach can effectively capture the semantic context factors of words, and it can be successfully used in IR in combination with the existing approaches.

The remainder of this paper is structured as follows: In Sect. 2 we present related work on WSD and the uses of word contexts and topic modeling techniques. In Sect. 3 we present our proposed latent context models. Section 4 describes how the context model may be used for ad-hoc search. Section 5 presents experimental results. In Sect. 6 we discuss the computational complexity of the technique, and it's impact when varying parameters such as the number of topics and word models. We finish with a discussion of our experiments and future research directions in Sect. 7 and we conclude in Sect. 8.

2 Related work

The technique presented in this paper combines IR, topic modeling, and WSD techniques. There is a vast literature on these topics. In this paper, we will only review the most related studies. For more complete overviews of WSD, readers are invited to read Navigli (2009) and Ide and Veronis (1998). For an overview of topic modeling, one can refer to Blei and Lafferty (2009), and on IR, to Croft et al. (2010) or Manning et al. (2008).

2.1 Words and topicality

In a typical IR scenario, while users formulate queries using specific words, they are actually interested in the concepts or topics implied by the keywords. So they usually expect that documents and queries could be matched using higher level features than words. For this purpose, latent semantic analysis (LSA) (Deerwester et al. 1990) has been proposed to convert the high-dimensionality word-space representations of the documents to low-dimensionality vectors of topics. Topics can be obtained using singular-value decomposition (SVD). Documents can then be ranked by their cosine similarity with the query's topic vectors, in the topic space. Using such a technique yielded some improvement over a tf-idf baseline. A closely related technique—pLSI (Hofmann 1999), tries to create a set of topics in a probabilistic framework. The topics in pLSA are probabilistic instead of the heuristic geometric distances in LSA. pLSA has been successfully used on large collections in IR, because its does not need to run the expensive SVD operation.

Latent Dirichlet allocation (LDA) (Blei et al. 2003) is a popular topic modeling tool designed to learn a set of topics (word distributions) and to infer mixtures of these topics to build low-dimensionality representations of documents. This model further refines the pLSI model (Hofmann 1999) within a Bayesian framework. LDA proposes the following generative process for documents (as shown in Fig. 1):

- Assuming K topics, for each topic of the model, pick a distribution from a Dirichlet with hyperparameter β.
  - $\phi_{k,w}$ is then the probability of the word w in the topic k.
- For each document d in the collection $\mathcal{C}$ (containing M documents)
  - Pick a multinomial distribution θ_d from a Dirichlet of hyperparameter α.
    - θ_d is then the topic mixture for document d.
  - For each token w_i, $i = 1..N_d$ of document d
    - Pick a topic z_i from the multinomial θ_d
    - Pick the token w_i from the multinomial ϕ_{z_i}

One advantage of LDA over traditional clustering techniques is that the Bayesian framework it uses allows for inferences to be made as a function of the quantity of observations rather than in a heuristic manner. Furthermore, individual words of the same document can belong to different topics. This is in contrast to regular term clustering techniques. For instance, in LDA, the likelihood of the word tokens w of document d, with regard to a set of topics and a document topic mixture is

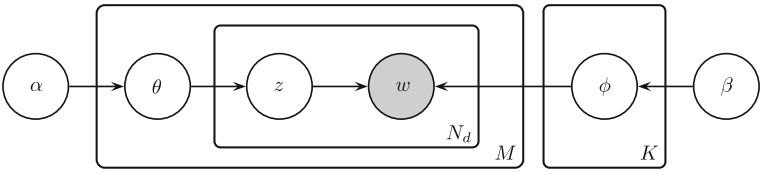

Fig. 1 Graphical model representation of LDA

$$P(w|\alpha,\phi,\theta_d) = P(\theta_d|\alpha)\prod_{i=1}^{N_d}\left[\sum_{k=1}^{K} P(w_i|z=k)P(z=k|\theta_d)\right]$$

As can be seen, the mixture of topics is determined in the inner sum. However, while the interactions between topics allow for a better modeling, it makes the approach intractable, and thus approximate inference methods must be used. We will use this model in this paper, but will leave the implementation details for later, in Sect. 3.2.

LDA has been used for IR in LDA document models (Wei and Croft 2006) as a tool to conduct document expansion: a set of topics are learned on the whole collection and a mixture of topic is inferred for each document in the collection. The traditional bag-of-word model is then smoothed using both the collection language model and the latent features extracted by LDA. Improvement in retrieval effectiveness has been observed. However, the more topics we use, the more computationally expensive the approach becomes. Conversely, a limited number of topics will result in a coarse topic granularity and in lower improvement. In practice, it is difficult to use this technique due to its high computational cost. Indeed, to compute the probability of a word in a document, one has to fetch the inferred topic mixture of the document and use it in a weighted sum of the probabilities of the word in the topics. This operation is computationally expensive. An interesting comparison of pLSA and LDA on several tasks (document clustering, classification and retrieval) can be found in Lu et al. (2011).

Hyperspace analogue to language (HAL) (Lund and Burgess 1996) is a context model that represents the context of a word by a vector of context words. It shows that word similarity can be computed by the cosine similarity of their context vectors. This has been applied, for instance, in Bai et al. (2005). An interesting difference in this method, compared with most other ones, is the implicit consideration of word distance: Rather than using a simple count of co-occurrences, the count of a context word is determined using a fixed-size sliding window over the text. The count of a context word is the number of such windows that contain both the target word and the context word. Implicitly, words that appear closer to the target word in the context will be counted more times. The sliding window amounts to a linearly decaying weight applied in function of distance.

2.2 WSD in IR

In this section we will present some relevant work on the use of WSD in IR.

Krovetz and Croft (1992) stated that word ambiguity can be divided into two categories: syntactic ambiguity and polysemy. The main difference between the two is that the former is due to different words sharing the same form, and the latter implies a semantic link. Given that keywords of search queries are often semantically cohesive, syntactic ambiguity would be less problematic in IR, since the intended word senses are likely to co-occur in documents. Krovetz and Croft (1992) also presented experiments showing that there is often a dominating sense (dictionary entry), in both document and queries, and thus, the intended sense of a keyword is often the most frequent in the collection, making ambiguity a non-issue for many keywords.

Sanderson (1994) further studied the impact of WSD on IR by randomly merging word forms (pseudo-words) to create artificial ambiguity. When testing IR effectiveness afterward, it was found that ambiguity has a minor effect on results. However, the merging of word forms was not made according to the Zipfian nature of language: the query words tend to be frequent words, while randomly selecting a word from the vocabulary may

equally favor rare words (Gaustad 2001). Stokoe (2005) later proposed an adaptation of the technique, which produces polysemic ambiguity by merging words which are related in the WordNet hierarchy. The added polysemy is shown to decrease retrieval effectiveness more than the random pseudo-words created by Sanderson. These studies present evidence of the presence of syntactic and polysemic ambiguity and show that the latter type may have worse consequences than previously thought.

WSD usually relies on manually created inventories and linguistic resources. This is known to be problematic because of coverage issues and the problem of determining the right granularity of an inventory (Kilgarriff 1997). As an alternative, Schütze (1998), Schutze and Pedersen (1995) proposed to cluster word contexts to build more specific indexing units. Each occurrence of a word is represented by a context vector using tf-idf weights. Clusters of context vectors are created, and a word occurrence is assigned to the most similar cluster to it. The resulting cluster assignments then become specialized indexing units. The clusters act as an inventory of senses. Assigning a word occurrence to such a cluster implicitly solve word ambiguity. Using this technique, Schütze and Pederson reported a relative increase of 14 % in average precision on a small set of 25 queries on the TREC WSJ collection. Sanderson (2000) later explained that the improvements could be attributed to the use of long queries containing tens of keywords. Thus a word in a query benefit from extensive context information, which is not the case for short queries.

Two recent studies have shown some positive results by using WSD techniques. Stokoe et al. (2003) postulates that the most important factor for achieving positive WSD effects is to manage the skewed sense distribution of words and minimize the impact of WSD failures when they occur. With a WSD system using WordNet as sense inventory and trained on the sense-tagged corpus SemCor, experiments on TREC topics 451–550 on the WT10G collection resulted in an average precision of 0.054 when using WSD, compared to a baseline of 0.034. However, these effectiveness levels are low compared to those in other studies (usually around 0.2 for this collection). It is unclear if the advantage of the approach can materialize on a stronger baseline method. A second study presenting positive results is Kim et al. (2004), in which, to increase the disambiguation accuracy, coarser senses corresponding to the broadest level of the WordNet sense hierarchy was used. Improvements were observed when using various tf-idf vector-space baseline ranking formulas. But again, the effectiveness is still not higher than what was obtained with the best BM25 baseline.

The above studies do not provide strong evidence that the direct use of traditional WSD techniques in IR could be effective. In addition, for an application with a large amount of data such as IR, a major problem of WSD is the need to have a sense inventory and an accurate disambiguation tool to determine the senses of words, which are far from being available. A more flexible method should be used.

2.3 Using local context in ranking

A less ambitious task to deal with ambiguity is to establish statistical constraints on context features, so that word occurrences with different meanings could be discriminated through these constraints. Indeed, occurrences of words within the same or a similar context would likely denote the same meaning, while those with different contexts would denote different meanings. Many IR techniques can be viewed as exploiting a similar principle. Passage retrieval (Salton et al. 1993) implicitly uses this idea: it requires that the query terms be contained in a small text segment (context) rather than within the whole document. This means that the query terms should appear in the same document context (passage), which

tends to correspond to that of the query. Term proximity (Croft et al. 2009) is another flexible criterion to capture the query term's mutual contextual influence. It has been used in several studies to determine the context (text span) in which the query terms occur. The smaller the text span, the stronger the correspondence between the document context and query context of the query terms. The positional language model for IR (Lv and Zhai 2009) is another approach trying to capture term proximity information. All these approaches share one common objective: to favor documents in which the query terms appear close together, i.e., in a context similar to the query. However, the way that context is defined differs largely. In particular, the ranking score is boosted according to term proximity, which is measured based on the distance between query terms in the document. The larger the distance is, the smaller the ranking score is boosted. Several manually defined decaying functions have been used in the previous studies. However, it is not clear how context words should impact the meaning of the target word, or how the context words at different distances should be weighted in a context model. This is the question investigated in Brosseau-Villeneuve et al. (2011), which we review in the next section.

2.4 WSD using optimally weighted context models

In many WSD and IR systems, all words within a text window of fixed size in documents (e.g., 8 words in Metzler and Croft 2005) around a target word are given equal importance to define the context. However, intuitively, the words closer to the target word have a stronger relation with the latter. HAL (Lund and Burgess 1996) uses a similar intuition. It defines the context of a target word by moving a sliding window of fixed size on the text. A context word is weighted according to the number of windows in which the context word co-occurs with the target word. Implicitly, the weight of a context word decays linearly along with the distance. In some proximity models (Lv and Zhai 2009), several decaying functions have also been defined and tested. It is expected that using a good decaying function, the weight assigned to a context word can better correspond to its influence on the meaning of the target word. However, there is no clear criterion (except the experimental results in IR) to choose the appropriate decaying function to use.

Rather than trying out all the possible decaying functions, which is impossible given the infinite number of such functions, we believe that the best way to find out the optimal weighting function is to make it emerge from the data. As such, we proposed to use unsupervised techniques to learn the function (Brosseau-Villeneuve et al. 2011) as follows: given a set of context samples of a word, we split this set randomly into two and we assume that the meanings of the target word in the two subsets to be similar (even if it is ambiguous in both subsets). Therefore, the context vectors extracted from the two subsets should be similar as well. So the optimal decaying function should be the one that maximizes the similarity of context models constructed from the two sets. More specifically, we randomly select a set of target words and obtain their context vectors. For each target word, the context vectors are randomly divided into two subsets. We then construct an aggregated context model for each subset, by assuming a weighting function $w(x)$ for the word at distance x with the target word, which is initialized as a uniform weighting. The similarity between the context models of the two subsets is determined their mutual cross-entropy

$$\mathsf{sim}(p, q) = H(p_{\mathsf{ML}}, q_{\mathsf{Dir}}) + H(q_{\mathsf{ML}}, p_{\mathsf{Dir}})$$

where to prevent null probabilities one of the two model is smoothed. Then the goal of unsupervised learning is to determine a weighting function such that the above similarity is maximized for all the target words t and their subsets context models A_t and B_t:

$$w^{\star} = \arg\max \sum_{t} sim(A_t, B_t)$$

This optimization problem was solved using gradient descent, which iteratively revises the weighting function so as to maximize the similarity of context vectors.

Figure 2 shows the weights for the 40 closest positions around the target word (100 positions were used) resulting from the learning process on an English corpus—English lexical sample (ELS) which is used in the Semeval-2007 experiments, together with several manually defined functions (uniform, linear, exponential and Gaussian) used in Lv and Zhai (2009). As can be seen from Fig. 2, the learned function is similar to a negative power law ($x^{-\delta}$ where x is the distance, and $\delta > 1$ a decay control parameter—$\delta = 1.1$ in the figure). We also plot the power law function in Figure 2, which had never been used in previous studies. The WSD experiments on Semeval-2007 ELS and Semeval-2010 Japanese WSD (Okumura et al. 2010) confirmed the superiority of the learned function and power law function (they perform equally well, as one may expect) over others proposed in the literature. WSD is performed by classifying the context vector of a test word occurrence into the sense classes modeled by the prototypical distribution of context words for the senses. Using the context vectors in which context words are weighted using our method, a Naïve Bayes classifier achieved state-of-the-art accuracy while using only simple co-occurrence statistics. On ELS, we obtained an accuracy of 0.8831, matching the best system in Semeval 2007 (0.8868) (Cai et al. 2007), which used many more sophisticated features (POS tagging, collocations, topic models and syntactic relations). In a Japanese WSD task, our system ranked the best at Semeval 2010 (Okumura et al. 2010).

Our WSD system has proven to be highly efficient and accurate, while being applicable to any language. It is thus a tool well adapted to large-scale WSD or WCD in IR tasks.

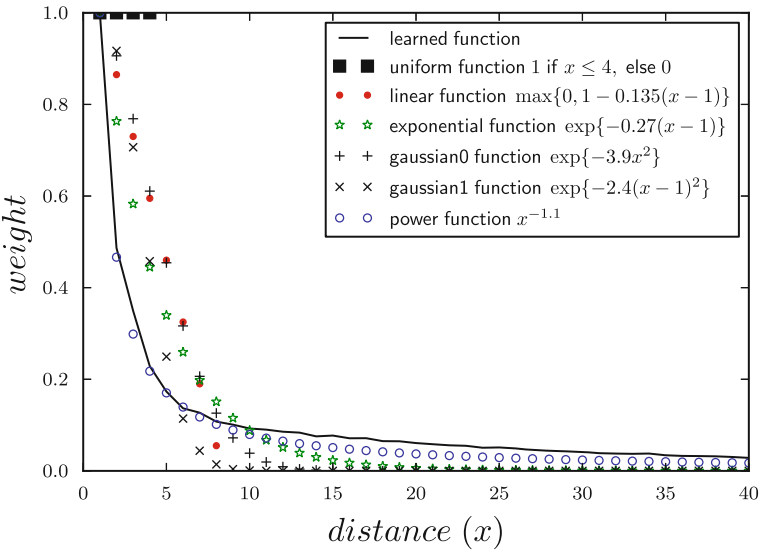

Fig. 2 Optimal settings for popular decaying function formulas and learned function on Semeval-2007 English lexical sample

3 Latent context models of words

As we mentioned earlier, a critical problem using a word-based context representation is its sparsity: similar contexts can be represented by different words. To relate similar context words, an approach is to create a more abstract representation in which similar words are grouped. Schütze (1998), Schutze and Pedersen (1995) used a similar idea by grouping similar context vectors into clusters. However, their approach also suffers from the word sparsity issue, as the similarity between context vectors is basically determined by how much the two vectors share common terms. Rather than a direct comparison based on words, we intend to create a more abstract representation in which similar words are naturally grouped into topics, and context vectors can be compared by on how much they share common topics. LDA is such a mechanism allowing us to do so. More specifically, we will create a topic model using LDA for each word. This choice, compared to the one that defines a unique LDA topic model for all the words, is motivated by the fact that different words have very different important topical aspects. We can obtain a finer granularity by using a separate topic model for each word. To do this, we first collect the contexts for each word, in which context words are weighted using the decaying function we determined before. Then an LDA topic model is created on the set of contexts of the word. One might believe that this will dramatically increase the complexity of the approach and make it unusable. We will show in Sect. 6 that this is not the case.

3.1 Context representation

Let us first examine the features to be used in contexts. In addition to content-bearing context words, as we suggested so far, there may be other features useful for WSD. Standard WSD systems use complex features such as POS-tags and syntactic dependencies. However, it is difficult to use these features in applications such as IR due to the heavy resources required. In addition, search queries are often ungrammatical, making it difficult to determine the precise syntactic information. Co-occurring word are simpler features which can be easily obtained. Given the fact that the simple features we used in WSD experiments resulted in high accuracy, we simply reuse the same featured in this paper. More precisely, for a target word, we will create a context vector containing the following features:

Target word form: The common use of stemming in IR increases recall, but at the cost of added ambiguity. For instance, the word "banking" may be conflated to the "bank" equivalence class. As we can see, the raw forms of a word often tend towards a particular sense (e.g., "banking" excludes the "river bank" sense). Therefore, the raw form is used as one of our features.

Neighboring stop words: While we can assume that most stop word instances in a window have poor semantic content, the stop words located right before or next to the target word tend to indicate a specific syntactic category of the target word. For example, "plant a" tends towards either the verb *to plant* while "a plant" tend towards either the *vegetation* or the *factory* senses. For more expressiveness, consecutive stop words are merged in a single feature, and stop words appearing before or after the target word correspond to different features.

Context content words: The content words appearing in the context of the target word are the main feature source for WSD systems. They are most often used as bag-of-words features within a small fixed window. Motivated by our WSD experiments, we replace this fixed window with a weighting function based on word distance. As we have seen previously, the optimal decaying function is close to the negative power law. Therefore, a context word at distance x will have the count $x^{-\delta}$, where δ is a parameter controlling the decay rate. In this paper, we simply use the value $\delta = 1.1$ determined in our WSD experiments. As in our previous WSD experiment, stop words are counted in distance, but consecutive stop words count as a single unit of distance.

To illustrate the feature scheme, the phrase "His plane **banked** to the right" translates to the features {*con*-2*_hi*, *con*-1*_plane*, *stop-left_*, *tar_banked*, *stop -right_to_the*, *con* -2*_right*}. The reader can notice that we stemmed "his" into "hi", and that the consecutive stop words "to the" were grouped in a single feature.

The context words *con-x* will be weighted by $x^{-\delta}$. One may notice that the decaying functions and the features we used in WSD may not be the ones that fit the best to IR. Indeed, in order to determine the optimal weighting function, we have to run the learning process on the document collection to be searched and to test the usefulness of different features for IR tasks. The training of a new weighting function on the document collection to be searched can indeed adapt the weighting function to the collection. However, in a realistic setting, the document collection constantly changes. It is unrealistic to get the optimal weighting function from the collection. Therefore, the weighting function is always trained on a (slightly) different collection. We simply use the weighting function trained on another collection (ELS) to simulate this situation. The three groups of features turned out to be all very useful in our WSD experiments. Ideally, we should test their usefulness and that of other features for IR. However, the main purpose of this paper is not to propose the best features to use, but rather an approach that uses a more abstract representation of contexts. We assume that contexts could be represented by any features that are easily available. The search for useful features will be left to our future study.

3.2 Building the models

The context models are built using the features stated above. Different from the previous uses of LDA in IR, we construct one LDA model for each word (stem) of the vocabulary. Context windows containing the word are accumulated, converted into context features with count/weight, those for the same document are aggregated into a single pseudo-document by adding counts for each window. This merging may cope with the sparsity issue and fits the common idea of *one sense per discourse* (Gale et al. 1992).

As the weights for context terms at large distances have very low weights, their impact on the meaning of the target word is small. Therefore, we limit context terms to a distance of 20. We assume that at least 500 documents are necessary to model a word's context, and that 100,000 documents containing the word are sufficient for model training. In other words, no context model will be constructed for words appearing in less than 500 documents, and for those that appear in more than 100,000 documents, only 100,000 documents are used to build their context models. Context words that appear rarely in the document

collection are not very useful features to discriminate word senses. Therefore, we kept only the features occurring at least 10 times. These measures help reduce the size of the models but will not significantly impact effectiveness.

To control the impact of each type of feature in the models, the counts can be multiplied by constant factors. For our experiments, we used the same values as we did in our supervised WSD experiments: $\omega_{\text{tar}}^{\text{model}} = 1.0$ controls the target word, $\omega_{\text{stop}}^{\text{model}} = 1.0$ controls the stop words, and $\omega_{\text{con},x}^{\text{model}} = 1.25x^{-\delta}$ with $\delta = 1.1$ (negative power law as a decaying function of distance x) controls the context content words. All the features share the same probability density, forming a single multinomial: we found that using distinct multinomials leads to coarse models, which did not yield good results in WSD.

The basic LDA model proposes the use of uniform Dirichlet priors on a topic's word distribution. In a typical use, there is a single uniform prior. When LDA is used on the whole collection, this is a reasonable choice. However, when different LDA models are created for different words, we can no longer assume that the same Dirichlet prior is shared by all the words. Intuitively, more frequent words would require a larger Dirichlet prior. Thus, we use the prior $\beta_{\text{con},x} = 1,000P(x|\mathcal{C})$ for the word x. For the target word forms and the stop word features, we used the standard uniform pseudo-count of $\beta_{\text{tar}} = \beta_{\text{stop}} = 1.0$.

We used a small constant number of topics for each word (K=10) by assuming that a word usually does not have more than 10 different meanings. A small K allows us to produce an efficient ranking procedure, as for each inverted index entry for one stem and document we will need to loop over all topics. The initial topic mixture hyperparameter α was set to 0.1 (sums to 1 with $K = 10$), and then computed using the fast Newton's method proposed by Blei et al. (2003).

For the inference method, we used Bayesian variational inference (VB) (Blei et al. 2003). VB is often considered to be slow because the digamma function needs to be computed for each topic. However, in our case this is not problematic since we use a small number of topics.

Starting from Blei's lda-c package source,[1] we added the support for real numbered counts, non-uniform priors on topics, and converted the LDA M phase (computing the topics ϕ) to an iterative update after a document's inference is recomputed. We selected the best of 5 random seeds by running three LDA iterations for each and keeping the seed producing the highest likelihood. We used lda-c's default convergence bounds. The resulting software has been combined with the Indri IR libraries[2] and is available for download[3]

4 Retrieval using latent context models

The traditional language model uses word unigrams to match a query with documents. Given a query Q, the score for a document d is defined by the following likelihood to generate the query by the document model:

[1] http://www.cs.princeton.edu/~blei/lda-c.

[2] http://www.lemurproject.org/indri.

[3] http://sourceforge.net/projects/latentcontext.

$$S_{\text{ql}}(Q,d) = \sum_{q \in Q} [W(q,Q) \log p(q|d)] \tag{1}$$

where $W(q, Q)$ is the weight of word q in the query, which is usually set as the frequency of the word, i.e. $W(q, Q) = n(q, Q)$

The document model is usually smoothed with the collection model. A common smoothing method is the following Dirichlet smoothing using a Dirichlet prior μ (Croft et al. 2010):

$$p(q|d) = \frac{n(q,d) + \mu p(q|\mathcal{C})}{\sum_x n(x,d) + \mu} \tag{2}$$

where $n(x, d)$ is the number of occurrences of word x in document d.

As we can see, the matching score is solely determined by the likelihood that the document model can generate the query terms separately. As we discussed, ambiguous query terms (e.g. "Java") could be matched with the terms in different meanings.

The LDA model we constructed for each word allows us to describe the word by a set of topics depending on the context words we can find around the word occurrence. In the ideal case, one would expect that the word "Java" could be described by different topics depending on what context words it is associated: program, computer... Indonesia, Bali, island... flavor, full-bodied, taste... If the query word "Java" is surrounded by similar context words as in a document, then the document is considered to described the same concept. Otherwise, the word would have different meanings in the query and in the document. We discussed earlier about the sparsity issue of context words. For the "Java" example, if the term "Java" is surrounded by "flavor", while the document's context only contains "taste", no match can be made according to context. However, semantically, the two contexts are similar. The LDA word context model we created tries to solve this problem. The context of a word is no longer described by words, but by the topics they represent. This allows us to perform the desired matching in the above example.

The LDA construction process we described in the last section will result in a set of models. Let $\phi_{w,k,f}$ be the probability of feature f given topic k of the context model for the word w, i.e., it represents the probability of the feature f in the kth topic of the word w. Notice that we use both f and w in this notation. w means a target word, while f means a context feature, which is one of the three types of feature we defined earlier: raw word form, stop words before and after the word and the context word stems. Running inferences for each collection document d containing w gives us parameters $\gamma_{w,d}$ (a set of K pseudo-counts) for a Dirichlet distribution of topics. The expectation of this distribution for the kth topic is as follows:

$$\theta_{w,d,k} = E\left[Dir(\gamma_{w,d})\right]_k = \frac{\gamma_{w,d,k}}{\sum_{k'}^{K} \gamma_{w,d,k'}}$$

which is the probability of the kth topic of the word w in document d. We use the above expectation to compute the probability of context feature f given word w and document d:

$$P(f|w,d) = \sum_{k=1}^{K} \phi_{w,k,f} \theta_{w,d,k}$$

We use the same features in queries as we do for documents. However there is a difference between modeling the context of a word in document, and assessing the relevance of documents. For instance, while stop word features introduce lexical and syntactic components in the models, the users often do not write them in queries. These features are

less important in IR than in WSD. We thus propose a different set of weights for the scoring component. Let f be a context feature, its weight in a query is defined as

$$\omega_f^{\text{query}} = \begin{cases} \omega_{\text{tar}}^{\text{query}} & \text{if } f \text{ is a target word form} \\ \omega_{\text{stop}}^{\text{query}} & \text{if } f \text{ is a stop word feature} \\ \omega_{\text{con},x}^{\text{query}} & \text{if } f \text{ is a content word at distance } x \\ 0 & \text{if } f \text{ is not in the model vocabulary} \end{cases}$$

where $\omega_{\text{con},x}^{\text{query}}$ is set in our experiments to the same power law as we used for documents, multiplied by the weight $\omega_{\text{con}}^{\text{query}}$ put on context word features:

$$\omega_{\text{con},x}^{\text{query}} = \omega_{\text{con}}^{\text{query}} x^{-\delta}, \delta = 1.1$$

The three parameters $\omega_{\text{tar}}^{\text{query}}, \omega_{\text{stop}}^{\text{query}}$ and $\omega_{\text{con}}^{\text{query}}$ will be tuned during the experiments (see Sect. 5.2.3).

Let F_q be the set of context features for keyword q in query Q. Using the ω_f^{query} weights above, we obtain the count $n(f, q)$ for feature $f \in F_q$. The log-likelihood of these features for a document/keyword pair is then:

$$\log p(F_q|q,d) = \sum_f n(f,q) \log p(f|q,d) = \sum_f n(f,q) \sum_{k=1}^{K} \phi_{q,k,f} \theta_{q,d,k}$$

The above quantity defines a new matching score based on context.

One may use the above score alone to rank documents. However, as shown in Lu et al. (2011), using an LDA alone as a ranking function results in poor retrieval effectiveness. It is better to combine it with a traditional language model. In this paper, we use a simple combination: the query likelihood and the query word's context likelihood are simply multiplied (i.e., add log-probs):

$$S_{\text{cm}}(Q|d) = S_{\text{ql}}(Q,d) + \sum_{q \in Q} [W(q,Q) \log p(F_q|q,d)] \quad (3)$$

where the relative importance of the features is controlled by the weighting parameters ω_f^{query} used in $p(F_q|q, d)$. The above formula means that a good document should match the words in the query (the ql part) and the context of query words in it should also match those in the query (second part in the formula). In this formula, we put equal importance on the two parts. It is possible to assign different importances on each component, and to tune the weights in experiments, and this could result in better retrieval effectiveness. We will leave this for our future study, and use the above simple form in this paper.

5 Experiments and results

5.1 Experimental setting

Our experiments were conducted on three TREC collections: AP 88-90, Robust 2004 and WT10G. We use topic titles as our short queries. The main characteristics of the collections are listed in the following table:

Collection	# docs	Size	# queries
AP 88-90 (*ap*)	242,918	767 MB	99
Robust 2004 (*robust*)	528,155	1.9 GB	248
(*robust-hard*)			50
WT10G (*wt10g*)	1,692,096	11.3 GB	98

The Robust 2004 collection queries contains a subset of hard queries (*robust-hard*) selected by the task organizers for their low score on many systems (Voorhees 2004). We will also show results for these queries, using a subset of the *robust* run (i.e., we do not choose parameters for these queries alone). The queries for the WT10G collection contains many terms that are not useful for IR, such as "Where can I find" in "Where can I find growth rates for pine trees?". We used simple patterns to remove these words from the queries (Appendix 2). The collections used inverted indexes of stems produced with the Porter stemmer, stopped at query time with the standard Indri stoplist, and we only used the title field of the topic descriptions.

5.2 Compared systems

The baseline query likelihood system (**ql**, Eq. 1) has one parameter μ to control the level of smoothing. Our proposed context model approach (**cm**, Eq. 3) adds the three parameters $\omega_{\text{tar}}^{\text{query}}$, $\omega_{\text{stop}}^{\text{query}}$ and $\omega_{\text{con}}^{\text{query}}$ to weight the new features. In addition to **ql** and **cm**, we will also compare our approach to two state-of-the-art approaches using blind relevance feedback and using positional index, namely, the relevance model and Markov Random Filed model.

5.2.1 Combining with blind feedback

Blind relevance feedback is a common approach to improve IR effectiveness, which exploits the top retrieved results to create a new query model / ranking function. Relevance model (Lavrenko and Croft 2001) is a typical example. This model is defined as follows. Let S_{used} be the score used for the initial query (for example, S_{ql}), Relevance model implemented in Indri defines the following scoring function:

$$S_{\text{rm}}(Q|d) = \lambda \frac{S_{used}(Q,d)}{\sum_{q \in Q} W(q,Q)} + (1-\lambda) \frac{S_{\text{ql}}(Q',d)}{\sum_{q \in Q'} W(q,Q')} \tag{4}$$

where λ is the weight of the original query score, and the expanded query Q' is defined as the set of feedback terms *fbTerms* with the biggest weights:

$$W(q,Q') = \sum_{D \in UsedDocs} S_{used}(Q|D) \frac{n(q,D)}{\sum_x n(x,D)}$$

where *UsedDocs* is the set of the *fbDocs* documents having the top $S_{used}(Q|D)$. As we can see in Eq. 4, the expanded query Q' is scored using the query likelihood formula (Eq. 1).

Relevance feedback in general has been proven to be a highly effective way to improve retrieval effectiveness. The technique can be combined with any basic retrieval model. Indeed, S_{used} in Eq. 4 can be replaced with any score function to be used in the first round retrieval. In our case, we can also use S_{cm} for it. Such a combination is reasonable in our

case as relevance feedback exploits a different aspect of IR from our approach. We will test how such a combination is effective, and whether the improvements brought by Relevance model and our model are additive.

5.2.2 Comparing to positional indexes

One well known way to increase retrieval effectiveness is to make use of positional indexes. We will thus compare our system with Metzler and Croft's MRF model (Metzler and Croft 2005). In this model, the relations between the document and the query words are modeled by an undirected graph G: the document and each query word are nodes, and edges between nodes are dependencies. The query words are combined by selecting cliques (fully connected subgraphs) of the graph. The following three graph structures are proposed:

full independence (fi):	Connections between query terms are not allowed. The features are thus the query's unigrams ($\|Q\|$ features).
sequential dependence (sd):	Only dependencies between neighboring query words are allowed. The features are thus the query's n-grams of arbitrary length ($\|Q\|(\|Q\|-1)$ features).
full dependence (fd):	Dependencies between any query word are allowed. The features are thus the query's skip-grams ($2^{\|Q\|}$ features). This third group is expensive to use on long queries.

From these cliques are formed three sets of features that must occur in the document: T is the set of *single terms*, O is the set of *ordered phrases* (n-grams preserving word order), and U is the set of *unordered phrases* (n-grams and skip-grams, ignoring word order).

Finally, given the features T, O and U extracted from a graph G_{sd} or G_{fd} for a query Q, with the weights λ_T, λ_O, λ_U, the score of document d is

$$S_{\mathsf{mrf}}(Q|d) = \frac{\lambda_T}{|T|}\sum_{f\in T}\log P(f|d) + \frac{\lambda_O}{|O|}\sum_{f\in O}\log P(f|d) + \frac{\lambda_U}{|U|}\sum_{f\in U}\log P(f|d) \qquad (5)$$

The probabilities of the features $P(f|d)$ is estimated from the number of occurrences of f in the document, smoothed with its probability in the collection. This is similar to the smoothing of unigram features of Eq. 2.

5.2.3 System settings

To compare our model with query expansion, we combine the two previous systems with relevance models (**rm**) in systems **ql+rm** and **cm+rm**, whose ranking are Eq. 4 where S_{used} is either **ql**'s (Eq. 1) or **cm**'s (Eq. 3). These two system add the additional parameters λ, *fbDocs*, and *fbTerms*. As this technique is costly (see Sect.6.2) and has many parameters, we limited the number of expansion terms to *fbTerms* $\leq$20 and the original query weight to under $\lambda \leq 0.7$.

Our model is also compared with the MRF ranking (Eq. 5) in the sequential dependency (**mrf-sd**) and full dependency (**mrf-fd**) variants. These two add the parameters λ_T, λ_O, λ_U to the basic query likelihood model. We used the script provided by the paper's authors,[4] which proposes using unordered window sizes of 4 times the number of words in the

[4] http://ciir.cs.umass.edu/~metzler/dm.pl.

feature (i.e. for a bigram, the window size is 8, while for a trigram, it is 12). We also tested the original fixed-sized window settings of size 2, 8 and 50, but observed that the variable setting worked best. So, we only report the results with variable window size.

For all systems, all parameters were selected by leave-one-out cross-validation, maximizing the metric of interest (e.g. MAP, nDCG, etc, which will be shown in the tables later). We performed a set of runs with in leave-one-our cross validation: We use grid search, to find the optimal setting for all but the test query. The range of values tested in grid search is as follows: μ: [1,000–4,000] with step 500, $\omega_{\text{tar}}^{\text{query}}$:[0.1, 0.6] step 0.05, $\omega_{\text{stop}}^{\text{query}}$:[0.05, 0.3] step 0.05, $\omega_{\text{con}}^{\text{query}}$:[0.1, 0.6] step 0.05, number of feedback documents *fbDocs*: [0–20] step 5, number of feedback terms *fbTerms*: [0–20] step 5, weight of original query (from Eq. 4) λ: [0.3–0.7] step 0.2.

6 Results

The retrieval results are listed in Table 1. In addition to MAP, nDCG and P@10, we also show the geometric means GMAP and GnDCG. By using a logarithmic scale to value effectiveness of individual queries, these latter measures focus on improvements on the harder queries, and when aggregated, evaluate the system's robustness.

The table contains experiments on each collection. Each collection has been processed using several retrieval methods: **ql** is the traditional baseline method. **mrf-sd** (sequential dependency) and **mrf-fd** (full dependency) are MRF models integrating both ordered and unordered groups of terms. These models are among the best performing methods in the literature, and they are also representative methods for using both bigrams and term proximity. **cm** is the context model we propose. **ql+rm** is a traditional method using blind relevance feedback on top of **ql**. **cm+rm** combines our method with blind relevance feedback in order to see if the improvements brought by both methods are complementary. The method we choose to compare to are the ones that consistently perform well on different test collections, and they represent the current state of the art.

There are several other methods based on term proximity that also perform well, but their effectiveness is roughly comparable to MRF models. So we do not include them here. We did not include other methods in our comparison because they have not produced state-of-the-art effectiveness in previous studies. In particular, as we discussed, the WCD method by Schütze and Pederson (1995) suffers from the sparsity problem in context representation. It can hardly be used to bridge the query context with the related document contexts that use different words. We do not include it in our experiments (Table 2).

When comparing **cm** with the **ql** baseline, we observe that our approach provides significant improvements on all collections. This strongly indicates the importance of taking the word context into account for IR.

When comparing **cm** with **mrf-sf** and **mrf-fd**, we see that **cm** is largely superior on *ap* and *robust*, and roughly equivalent on *wt10g*. However, as we will see in Sect. 6.4, the improvements of **cm** can be increased by an additional 50 % when using more topics. We thus conclude that our model is consistently superior to MRF. Also, we observe larger improvements than what was observed with other approaches using term proximity (Zhao and Yun 2009) and positions (Lv and Zhai 2009).

When comparing **cm** with **ql+rm**, we first observe that the latter generally performs better on MAP and nDCG. However, when we look at GMAP, GnDCG and P@10, we see that context models are generally superior. In the case of the harder collections (*robust-*

Table 1 Retrieval results with baseline system, latent context models and relevance model

Collection and run	MAP	GMAP	nDCG	GnDCG	P@10
ap ql	0.2220	0.0976	0.4745	0.3531	0.4273
ap mrf-sd	0.2308† (+4.01 %)	0.1049‡ (+7.45 %)	0.4849† (+1.60 %)	0.3887† (+10.74 %)	0.4293 (−0.23 %)
ap mrf-fd	0.2291 (+3.22 %)	0.1102† (+12.85 %)	0.4849 (+1.59 %)	0.3966 (+12.98 %)	0.4556‡ (+5.87 %)
ap cm	0.2472‡ (+11.35 %)	0.1188* (+21.68 %)	0.5102‡ (+7.52 %)	0.4161† (+17.83 %)	0.4616 (+8.04 %)
ap ql+rm	0.2699* (+21.60 %)	0.1211 (+24.04 %)	0.5554‡ (+12.98 %)	0.4051 (+14.72 %)	0.4747 (+11.11 %)
ap cm+rm	0.3082‡ (+38.83 %)	0.1557‡ (+59.54 %)	0.5786‡ (+21.94 %)	0.4757‡ (+59.54 %)	0.5202‡ (+21.94 %)
robust ql	0.2499	0.1390	0.5231	0.4534	0.4234
robust mrf-sd	0.2619‡ (+4.79 %)	0.1535‡ (+10.40 %)	0.5360‡ (+2.46 %)	0.4676‡ (+3.13 %)	0.4415† (+4.28 %)
robust mrf-fd	0.2629 (+5.20 %)	0.1555* (+11.90 %)	0.5360 (+2.47 %)	0.4595‡ (+1.34 %)	0.4500* (+6.28 %)
robust cm	0.2807‡ (+12.32 %)	0.1667‡ (+19.94 %)	0.5622‡ (+7.47 %)	0.4949‡ (+9.17 %)	0.4560 (+7.70 %)
robust ql+rm	0.2922* (+16.92 %)	0.1480‡ (+6.46 %)	0.5615 (+7.33 %)	0.4670‡ (+3.00 %)	0.4379* (+3.42 %)
robust cm+rm	0.3136‡ (+25.47 %)	0.1661‡ (+19.53 %)	0.5887‡ (+12.55 %)	0.5007‡ (+10.42 %)	0.4609† (+8.85 %)
robust-hard ql	0.0976	0.0577	0.3492	0.3047	0.2540
robust-hard mrf-sd	0.1058* (+8.44 %)	0.0648* (+12.33 %)	0.3620* (+3.67 %)	0.3148 (+3.33 %)	0.2700 (+6.30 %)
robust-hard mrf-fd	0.1048 (+7.35 %)	0.0649 (+12.46 %)	0.3632 (+4.01 %)	0.3075† (+0.91 %)	0.2780 (+9.45 %)
robust-hard cm	0.1132* (+16.01 %)	0.0714 (+23.58 %)	0.3918† (+12.21 %)	0.3400‡ (+11.59 %)	0.2800 (+10.24 %)
robust-hard ql+rm	0.1051 (+7.67 %)	0.0497† (−13.91 %)	0.3669 (+5.05 %)	0.2969* (−2.55 %)	0.2380* (−6.23 %)
robust-hard cm+rm	0.1219* (+24.87 %)	0.0598 (+3.68 %)	0.4029† (+15.38 %)	0.3272* (+7.38 %)	0.2780 (+9.45 %)
wt10g ql	0.1874	0.0740	0.4571	0.2879	0.2949
wt10g mrf-sd	0.2012* (+7.36 %)	0.0873† (+17.88 %)	0.4774‡ (+4.44 %)	0.3355‡ (+16.55 %)	0.3031 (+2.77 %)
wt10g mrf-fd	0.2048* (+9.33 %)	0.0874 (+18.02 %)	0.4736 (+3.60 %)	0.3288 (+14.20 %)	0.2990 (+1.38 %)
wt10g cm	0.2041 (+8.95 %)	0.0847 (+14.48 %)	0.4849* (+6.07 %)	0.3081 (+7.02 %)	0.3051 (+3.46 %)
wt10g ql+rm	0.1946 (+3.85 %)	0.0718‡ (−2.97 %)	0.4578‡ (+0.15 %)	0.2869† (−0.35 %)	0.3061 (+3.81 %)
wt10g cm+rm	0.2211† (+17.98 %)	0.0713 (−3.72 %)	0.4842* (+5.92 %)	0.3002 (+4.29 %)	0.3276 (+11.07 %)

Parameters are selected by leave-one-out cross-validation maximizing the column's metric. *robust-hard* uses the same parameters as *robust*

* $p < 0.05$; † $p < 0.01$; ‡ $p < 0.001$ (20M pass two-tailed randomization test with tie-splitting over the previous line system)

Table 2 Retrieval results when using 20 latent topics

Collection and run	MAP	GMAP	nDCG	GnDCG	P@10
ap cm (k = 20)	0.2525 (+13.75 %)	0.1225 (+25.48 %)	0.5143 (+ 8.40 %)	0.4207 (+20.20 %)	0.4889 (+14.42 %)
robust cm (k = 20)	0.2848 (+13.94 %)	0.1693 (+20.36 %)	0.5646 (+ 7.95 %)	0.4984 (+ 9.93 %)	0.4645 (+ 9.71 %)
robust-hard cm (k = 20)	0.1183 (+21.25 %)	0.0710 (+22.89 %)	0.3948 (+13.07 %)	0.3494 (+14.68 %)	0.3100 (+22.05 %)
wt10g cm (k = 20)	0.2119 (+13.11 %)	0.0921 (+24.48 %)	0.4930 (+ 7.84 %)	0.3387 (+17.65 %)	0.3153 (+ 6.92 %)

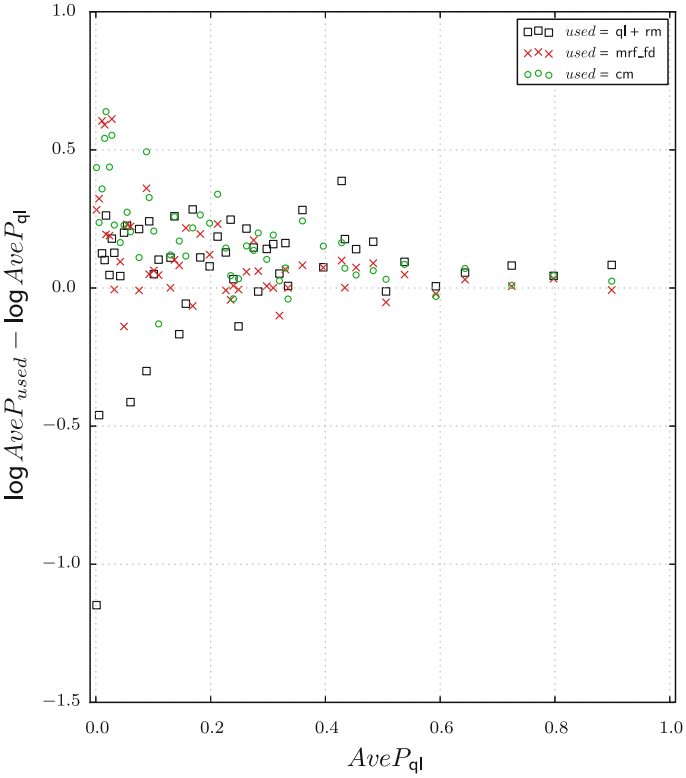

Fig. 3 Differences in *AveP* between systems and the query likelihood baseline on *robust-all* queries, in function of the query likelihood *AveP*. To improve readability consecutive data points were averaged in groups of five

hard, *wt10g*), we observe that **cm** is superior to **ql+rm** on all metrics, and that **ql+rm** is even harmful on many. This is to be expected with blind feedback techniques: on difficult queries, the top documents of the initial run contains few relevant documents, leading to poor expansion terms.

To confirm the instability of **ql+rm** and to have a general insight into the nature of the improvements of **cm**, we plotted differences in log-AveP (used in the geometric mean) with the baseline **ql** (Fig. 3) of the tree types of models. As can be seen, the improvements

Table 3 *ap* collection queries with worst decrease and best improvement when using context models

Changes in *AveP*
−0.1308 Military Coups Detat
−0.1160 Hostage Taking
−0.0887 Negotiating an End to the Nicaraguan Civil War
−0.0624 Capacity of the US Cellular Telephone Network
−0.0539 Israeli Role in Iran Contra Affair
0.1526 Surrogate Motherhood
0.1649 Impact of the 1986 Immigration Law
0.1925 US Political Campaign Financing
0.2191 Attempts to Revive the SALT II Treaty
0.2508 What Backing Does the National Rifle Association Have
Changes in log (*AveP*)
−1.3863 Politically Motivated Civil Disturbances
−0.9480 Demographic Shifts in the US
−0.7640 Hostage Taking
−0.5390 Computer aided Crime Detection
−0.5140 Military Coups Detat
0.7822 Attempts to Revive the SALT II Treaty
1.0591 Privatization of State Assets
1.1447 US USSR Arms Control Agreements
2.0714 Management Problems at the United Nations
4.9416 Black Monday

in log space generally comes from the harder queries (to the left) on which, **ql+rm** performs badly. We can also observe that **cm** and **mrf** have similar profiles, but **cm** is generally more consistent, while **mrf** brings the top improvements on single queries. Three cases stand out for **mrf**, they correspond to the queries "home schooling", "price fixing" and "World Court", which are (not surprisingly) idiomatic expressions. **mrf** performs very well for these queries because it integrates an ordered n-gram component which requires that the terms appear in the documents in exactly the same form as a phrase. This component fits well idiomatic expressions.

Comparing **cm+rm** and **ql+rm**, we observe that on metrics where **ql+rm** produces improvements, the **cm+rm** gives improvements even larger than the sum of those from **cm** and **ql+rm**. This clearly shows that **cm** and **rm** are complementary, and that our method can be combined with the existing methods based on blind relevance feedback. Our interpretation of this complementarity is that **cm** is a constrictive technique that improves precision (much like **mrf**), while **rm** is an expansive technique which improves recall. Using the former method, we can improve the quality of top retrieved documents in the first round, which in turn helps the latter select better expansion terms for a second-round retrieval.

To understand the nature of the queries where context models are effective, we listed the titles of the queries that had the best improvements and worst decreases (Table 3). As can be seen, the top increases came from ambiguous and/or collocational expressions in queries. In these cases, it is useful to enhance the matching between query and document contexts. However, in some other cases, the intended context is not well expressed in the query. Emphasizing too much on the context matching between document and query can harm the

retrieval effectiveness. This is the case for the queries with the largest decreases. For instance, the query "Hostage taking" intends to retrieve documents on "hostage taking for political reasons", while most of the retrieved documents discuss events occurring in prisons. As could be expected, our technique has the effect of further constricting the query scope, making the retrieval process more dependent on the query formulation. Our technique is less useful when the query is a partial or incorrect formulation of the information need.

7 Complexity

Complexity is an important issue when using topic modeling techniques. In this paper, we construct an LDA model for each term rather than for a whole document collection. While this would need more processing time and space than for computing one large LDA model, the technique has important advantages as we will show in this section.

7.1 Model building

As previously discussed, since we can assume that sufficiently accurate models can be made using a limited amount of data, the computational cost does not grow with the collection size. The model for one word is quite small and can be constructed quickly on a single machine. The cost of making inferences on new indexed documents is small considering the other costs involved in the indexing. Contrary to traditional LDA models, our models are easy to create and update when needed. For instance, a new LDA model can be added when a word becomes frequent enough to create an LDA model for it, or when new document contexts start to have low likelihood (e.g., when a new sense for a word emerges). For the words whose meaning remains unchanged, their context models do not need to be modified. The word context models can thus be modified incrementally.

7.2 Scoring

Techniques using topic models for IR such as LDA document models are difficult to use on large-scale collections because of their cost for evaluating queries. Indeed, the improvement shown in Wei and Croft (2006) was obtained as a high computational cost: all documents of the collection are involved in the scoring of one query. Although we can have an implementation limiting candidate documents to those containing one of the keywords, it would still take $\Theta(K|Q|)$ atomic operations per document to compute the keyword's probabilities. For instance, with the $K = 800$ topics (for small collections) as proposed in Wei and Croft (2006) and a query of three keywords ($|Q| = 3$), 4,800 operations per document[5] are required in addition to the regular index treatment. Our method uses extended index entries (no need to use all documents), resulting in $\Theta(K|F_q|)$ operations per used index posting (F_q being the set of context features for keyword q). As in our case $K = 10$ is very small and for the same query of three keywords there would be a maximum of 13 context features[6] (with all context features in vocabulary), we will have a

[5] 3 keywords × 800 topics × (1 add. + 1 mult. per topic) = 4,800.

[6] A query $Q = \{q_1, q_2, q_3\}$ made from three content words results in one target word feature and two context word features per keyword, and four "no stop word" stop word features (at $q_{1,right}$, $q_{2,left}$, $q_{2,right}$, $q_{3,left}$).

maximum of 260 operations per document[7] (when all keywords are in the document). This is a pessimistic upper bound. In practice unrelated query keyword features would not be in the model vocabulary, and often not all keywords are present in one document. To show the actual efficiency of different methods, we list the run times on the *wt10g* collection on with the best runs settings for MAP:

Run	ql	cm	mrf-sd	mrf-fd	ql+rm	cm+rm
Run time (s)	22	44	57	94	1,196	1,338

The **ql+rm** and **cm+rm** run uses $fbDocs = 10, fbTerms = 20$. As can be seen, the cost of context models is small compared to query expansion techniques such as relevance models. These run times are competitive with those of positional indexes, and have the advantage that they do not depend on the number of instances of a word in a document, nor involve iterating multiple indexes for complex cliques, resulting in reasonable scoring times. In comparison with additional terms from query expansion, the computation of context feature scores has good locality (i.e. the computation iterates on a small number of elements, which can be stored in cache), and does not introduce new documents in the results. We note, however, that the cost is greater on long queries, as the number of context features $|F_Q|$ grows in $O(|Q|^2)$. When the number of context features is large, we may consider pruning them (e.g., keeping the closest), or create approximations of the context feature's scoring component.

7.3 Number of used models

Another possible critics on our model is that it may seem daunting to build one model per word. To minimize the computational needs, we can exploit the Zipfian nature of language and create only models for the most frequent terms, maximizing the use of the built models. This pruning will not have large negative consequences, since rare words are very specialized, and thus less ambiguous. Hence using the technique for them would be less beneficial.

We have conducted experiments varying the number of used models. There, we assumed that words more frequent in the collection are also frequent in queries, and that using models for more frequent words will maximize the use of the models. We'll call this usage measure *query coverage* – the proportion of words in a query on which models are applied. When no word context model is used, the model reverts to the **ql** system. We selected the *robust*collection for this experiment for its large number of queries. Table 4 shows the results. As can be seen, following Zipf's law, the query coverage grows much faster than the number of models. For instance, for the runs of Table 1, we used a minimum collection frequency of 500 as a threshold. According to our estimate, this results in 15,429 models (approximately 2 % of the vocabulary) with a query coverage of 90.68 %. We can also observe in the last lines that limiting the number of word models slightly increases retrieval effectiveness. A possible reason for this may be the limited size of this particular collection, for which accurate models cannot be built for rare words. Conversely, like previously mentioned, rare words may tend to be highly specialized and thus not

[7] 13 features × 10 topics × (1 add. + 1 mult. per topic) = 260.

Table 4 Results on *robust* queries when varying the number of word models

Num. of models	Min. coll. freq.	Query coverage (%)	MAP	GMAP	nDCG	GnDCG	P@10
0		0.00	0.2499	0.1390	0.5230	0.4534	0.4234
100	273,764	3.85	0.2505 +0.21 %	0.1403 +0.91 %	0.5237 +0.14 %	0.4538 +0.09 %	0.4238 +0.10 %
200	160,898	7.84	0.2522 +0.89 %	0.1438 +3.41 %	0.5249 +3.68 %	0.4576 +0.92 %	0.4290 +1.33 %
500	80,327	18.20	0.2597 +3.90 %	0.1499 +7.78 %	0.5335 +2.01 %	0.4668 +2.96 %	0.4387 +3.62 %
1,000	40,235	35.06	0.2658 +6.36 %	0.1542 +10.89 %	0.5408 +3.39 %	0.4756 +4.89 %	0.4351 +2.76 %
2,000	15,239	54.59	0.2653 +6.15 %	0.1612 +15.93 %	0.5440 +4.02 %	0.4841 +6.77 %	0.4500 +6.29 %
5,000	3,339	76.78	0.2734 +9.38 %	0.1664 +19.70 %	0.5592 +6.92 %	0.4919 +8.50 %	0.4532 +7.05 %
10,000	1,088	86.39	0.2786 +11.48 %	0.1647 +18.46 %	0.5628 +7.61 %	0.4937 +8.90 %	0.4504 +6.38 %
20,000	333	93.05	0.2786 +11.46 %	0.1669 +20.02 %	0.5618 +7.41 %	0.4939 +8.95 %	0.4569 +7.90 %
50,000	62	97.34	0.2822 +12.92 %	0.1670 +20.14 %	0.5627 +7.59 %	0.4949 +9.16 %	0.4573 +8.00 %
100,000	21	99.11	0.2821 +12.86 %	0.1670 +20.09 %	0.5625 +7.55 %	0.4953 +9.25 %	0.4484 +5.90 %
771,975	1	100.00	0.2821 +12.86 %	0.1670 +20.08 %	0.5625 +7.55 %	0.4953 +9.25 %	0.4484 +5.90 %

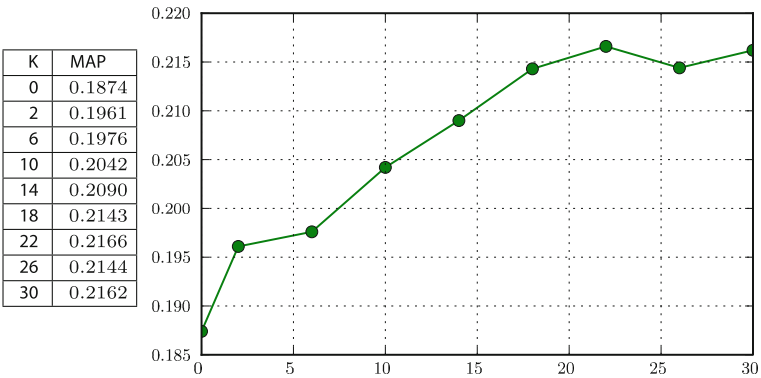

Fig. 4 Mean average precision on *wt10g* queries when varying topic count *K*

ambiguous. Introducing an unnecessary disambiguation mechanism for them may is useless and may even be harmful.

7.4 Number of used topics

For our main runs, we used 10 topics, assuming that the number of actual senses of a word does not exceed 10. It is interesting to see what would be the performance when using

more or less topics. Figure 4 lists MAP as a function of the number of topics on the *wt10g* collection. As we can see, the the number of topics has a high impact on the retrieval effectiveness. As could be expected, the step with the largest increase is the one going from the baseline ($K = 0$) towards a model with two topics. After this, the effectiveness increases somewhat linearly, and then stabilizes around $K = 20$. Although an increase in the number of topics leads to a higher effectiveness, one should also notice that this is obtained at the cost of a higher complexity. Therefore, a compromise should be made in practice. As mentioned previously, as context models can be made with a subset of the data, the cost of creating the models does not grow with collections size. On massive collections, the model's creation cost becomes small, and users may then consider optimizing the number of topics on a per-word basis rather than using the same for all words.

Given this new knowledge of the best number of topics for *wt10g*, we did a second set of runs using $K = 20$ topics (Table 2). The results are generally better with this setting, but with lesser improvements on *robust* compared to *wt10g*. One explanation is that with few topics, the models are more general (closer to word senses), while with many topics, they are closely fitting the underlying co-occurrences. Using more topics may thus be better when queries do not convey the query intent well. For instance, the *wt10g* topics were selected from web logs, and often the description and title do not match.

8 Future research

Using the proposed technique, we observed consistent improvements on all used test collections. The experiments demonstrated that our technique of WCD is effective for IR. Compared to the previous research results, the improvements we obtained are much larger. However, we made several simplifications in our implementation in this paper, which can be further investigated in the future.

In this study, we used the same parameters (decaying weighting function) learnt on a collection for word sense disambiguation, and applied them to a new collection in the same language and for a different (IR) task. This simplification is made because our primary goal of this paper is to show that our technique based on word context topic models can be effective and practically feasible for IR. We did not intend to find the optimal parameters for the IR task. This is one of the aspects we will investigate in the future.

To construct context models for query words, we used the same approach as for documents. However, the differences between queries and documents should be taken into account in a better processing. In this study, we have assumed that various types of features have different importance in documents and queries, and we thus used a leave-one-out method to tune these weights. This only account for some of the differences between queries and documents. Another important difference we need to cope with is the difference in length and in richness of contexts. As queries in practice are usually short (a few words), it is unreasonable to use a decaying function learnt on documents that spans over 100 words. A new decaying function specific for queries could be learnt. One possible way to do it is to train a new weighting function using a large set of queries, in a similar way as what we did on documents. This could be done using query logs. Another interesting avenue is to exploit clickthrough data in a similar way as learning-to-rank, but to learn the weights of context words: we try to assign weights to context words so as to maximize the final retrieval measure (e.g. MAP or nDCG).

In this study, we investigated two following aspects together: the creation of a word context using a decaying function, and the creation of a latent topic representation for word

contexts. We have not investigated the impact of each of them. It would be useful to do it in the future in order to understand the impact of each aspect. For example, what would be the retrieval effectiveness if we use decaying weighted context vectors, but without an LDA topic model, or with an LDA topic model on uniformed weighted context vectors? What is the impact of using LDA topic model instead of a more traditional clustering technique as the one used by Schütze and Pederson (1995)? All these questions need further investigations in the future.

Another interesting extension of this work would be to investigate the use of document fields in the construction of contexts. In our experiments, we merged the body and titles of documents without delimiting contexts within them. Some contexts can thus group words in different fields, which is unreasonable. A better way to extract contexts is to exploit the document structure so that contexts are limited in the same field. This strategy could be further used on different sections or paragraphs in the document body.

In this paper, we used the original LDA model for context modeling. Several extensions proposed during the recent years could also be used. For instance, Dirichlet compound multinomials have been used in topic models (Doyle and Elkan 2009) and they can better cope withword burstiness. As we can assume that word contexts are highly bursty, this may improve effectiveness. More sophisticated network models were also proposed, including Pachinko allocation (Li and Mccallum 2006) and Correlated topic models (Blei and Lafferty 2006). These models have been shown to be able to account for the relations between topics, producing better models using fewer topics. This is an interesting avenue for our future work because we have to limit the number of topics to a small number to gain efficiency in practice.

While the technique proposed in this paper mostly increases precision, it would also be useful to increase recall by creating query expansion techniques using the context models. Context models can indeed be seen as a flexible distribution of related words. Since they already include inference mechanics, it would be possible to use them to introduce dependencies between keywords, producing conditionally dependent related words for expansion.

The context models as defined in this study have the effect of constricting query scopes, reducing the variety of the results. Indeed, we score higher the mixture of topics in documents that maximized he likelihood of the query. In reality, the topics created for a word can correspond to the different interpretations of the word. This give us the possibility to exploit the context topic models in a different way to account for the possible multiple meanings of a query word: The multiple topics for the contexts of a word (or a query) can be considered as describing different intents behind the word. Each of such topic can be used in turn to retrieve a group of documents corresponding to the intent. By using results matching different topics, we can arrive at a new strategy for search result diversification.

Finally, we have limited ourselves to the simplest features for word context in order to gain higher efficiency. The features to be used can be further investigated. New features could be added; the features could go through a selection process in order to keep only some important ones, thereby the efficiency could be further improved; the number of topics could depend on the word—more topics could be created for words with more meanings;... All those aspects will help us better understand the impact of contexts on IR and the approach we propose in this paper.

9 Conclusion

Bag-of-words approaches to IR are recognized as being insufficient. Possible improvements to these approaches can be obtained by exploiting a stricter set of criteria for

document-query matching. In previous studies, the additional criteria are based on term dependencies, term positions and proximity. All these approaches have lead to improved retrieval effectiveness. In this paper, we exploit a different criterion—the matching of document context and query context for the query terms. Our assumption is that a term occurrence in a relevant document should denote the same meaning as that in the query, and thus their contexts in the document and in the query should be similar. Two problems have to be solved to implement this idea: (1) the construction of a context model for a term; and (2) the matching between two contexts. Our intuition for constructing the context model for a term is that a a context term closer to the target term has a greater impact on the meaning of the latter. Therefore, a decaying function on the distance should be used for weighting. For this reason, we conducted WSD experiments aiming to learn the optimal function using unsupervised methods. The experiments demonstrated that such weighted contexts can better capture the meaning of words than a uniform weighting. We then use the same weighting strategy for IR. This is the first time that such a decaying weighting function learned from data is used for IR.

Context matching could be made directly on the context models based on words. However, doing so, we would encounter sparsity issues, i.e. semantically similar contexts can be described by different words. We therefore used LDA to extract a set of topics to represent the context, leading to a latent semantic modeling of the context. Context matching based on the topics can be made at a more abstract level, alleviating the sparsity issue.

The proposed approach has been tested on several TREC collections. We observed substantial and significant improvements over the traditional approach and the state-of-the-art approaches using groups of terms and term proximity (MRF models). This strongly indicates that taking the context into consideration is crucial in IR and our method to do it is advantageous compared to the previous ones. Our experiments also showed that our approach is complementary to those using relevance feedback, and it can be combined with the latter to produce even larger improvements.

This study proposed a new method to model word contexts and to use them in IR. However, we have not fully exploited the potential of the approach. Simple implementation choices have been made. As we discussed in Sect. 6, many questions remains and further investigation is needed to better understand how contexts should be modeled and used in IR.

Acknowledgments The authors wants to thank Daniel Ramage and Takahiro Takasu for helpful comments. This work is partially supported by Japanese MEXT Grant-in-Aid for Scientific Research on Infoplosion (# 21013046) and the Japanese MEXT Research Student Scholarship program.

Appendix 1: A look in the topics

In this section, we'll try to have a look at what the context model topics represent. One way to do so is to list the top words of each topic. Rather than using the simple word frequency, we'll use a tf-idf-esque weighting proposed by Blei and Lafferty (2006): using the odds of the probability in the topics and the geometric mean of the probability over all topics (Table 5).

$$P(x|t_i)\log\frac{P(x|t_i)}{\prod_{j=1}^{T}p(x|t_j)^{1/T}} = P(x|t_i)\left(\log P(x|t_i) - \frac{1}{T}\sum_{j=1}^{T}\log p(x|t_j)\right)$$

The following sections show the first 8 topics made on the AP88-90 collection.

Bank

"west bank"	"river bank"	"development"	"japanese bank"	"investment banking"	"banking and government"	"bank robbery"	"bank account"
west	>of-the	world	>of	*banking	*banking	robberi	account
gaza	blood	loan	central	>of	board	<a	reserv
occupi	river	debt	japan	invest	committe	robber	swiss
>and	<on-the	creditor	intervent	corp	hous	food	fraud
strip	panama	commerci	tokyo	york	loan	<the	>of
palestinian	local	develop	dollar	new	senat	*bank	vault
isra	miami	intern	dealer	largest	save	rob	feder
villag	outer	financ	nation	chemic	home	<in-the	<in
town	panamanian	imf	republ	irv	regul	employe	cash
arab	american	>to	<the	compani	chairman	teller	borrow
israel	reopen	*bank	trader	firm	feder	>	travel
<	their	foreign	>in	merchant	>	branch	secreci
citi	close	<the	yen	deutsch	subcommitte	polic	non
nablu	piggi	privat	mitsui	manhattan	silverado	card	crete

Price

"price level"			"gas price"	"market price"		"oil price"	
*priced	support	futur	gasolin	stock	index	oil	bid
median	wheat	soybean	ga	bond	consum	crude	gold
home	farm	wheat	retail	share	wholesal	opec	>of
purchas	farmer	grain	averag	close	percent	barrel	bullion
low	milk	corn	fisher	fell	food	petroleum	silver
sale	corn	were	wholesal	tokyo	produc	stabil	late
rang	commod	mix	goug	todai	rose	have	recommend
bargain	crop	cattl	land	rose	0	world	ounc
base	higher	fell	pump	movement	energi	cartel	troi
high	market	gold	gallon	treasuri	increas	<the	zurich
hous	target	copper	per	dollar	y	iraq	london
000	>for	close	cent	market	inflat	product	rose
car	agricultur	chicago	unlead	yen	report	collaps	wa
highest	subsid	pork	natur	trade	veget	committe	fell
undisclos	averag	coffe	plung	finish	depart	benchmark	dealer

Product

“agriculture”	“industrial”	“petroleum”	“national”	“uclear”			“agriculture”
*product	*production	*production	*product	*production	*product	*productivity	*production
food	truck	oil	nation	weapon	*productive	*production	crop
agricultur	week	opec	gross	compani	blood	industri	wheat
dairi	schedul	quota	refin	televis	<the	increas	agricultur
tobacco	plant	barrel	petroleum	nuclear	<a	>and	soybean
such	at	crude	trade	film	>that	<in	milk
farm	>for-the	ceil	steel	>of-the	their	improv	grain
other	coal	<and	grew	facil	our	cost	corn
us	car	cut	domest	reactor	new	percent	dairi
meat	worker	cartel	percent	resum	we	0	estim
their	line	explor	rose	plutonium	develop	rose	farm
label	chrysler	limit	gasolin	movi	or	cutback	livestock
japanes	goe	price	among	materi	>and	growth	>of
ar	light	ga	>the	<and	tamper	<of	pork
appl	>of-the	increas	<	tv	qualiti	lost	year

Politic

*politic	reform	<a	<for	hi	*politically	<and	parti
<in	*political	*political	asylum	career	motiv	>and	*political
>and	system	action	or	*political	sensit	econom	opposit
>the	chang	settlement	*political	analyst	crisi	*political	leader
>for	plural	solut	reason	figur	influenc	turmoil	group
*polite	soviet	process	persecut	oppon	<that	social	<
>in	situat	contribut	spectrum	futur	realiti	pressur	wing
<	power	committe	refuge	violenc	favorit	stabil	organ
>of	gorbachev	climat	their	<	connect	instabl	appointe
nation	econom	donat	purpos	<in	embarrass	militari	movement
american	upheav	>	belief	power	consider	problem	forc
*politely	<the	<in-the	seek	life	gain	uncertainti	legal
he	structur	<the	view	her	expedi	clout	rival
plai	union	<of-a	flee	foe	wa	cultur	independ
partisan	communist	<for-a	grant	comeback	be	tension	ralli

Black

"black market"		"black monday"				"black object"	"colour"
>	*black	*black	*black	*black	*black	*black	*black
*black	>	>	>	>	>	>	>
<	<	<	<	<	<	<	<
market	township	mondai	first	percent	is	<the	>and
decker	were	>said	mayor	>and	<a	smoke	white
out	sea	church	elect	student	<of	hill	with
<on-the	<of	cathol	is	ar	he	<of	<and
*blacked	polic	<of	vote	<of	peopl	with	<in
<the	have	it	voter	white	<the	hole	<a
singl	faction	<on	he	among	it	bear	wear
own	kill	>a	<a	school	american	box	<in-a
by	black	is	who	hispan	who	wa	hi
it	<the	<in-the	citi	<for	as	<a	dress
lung	<in	wa	democrat	women	wa	from	red
clint	<in-the	bishop	candid	were	black	it	wa

Monday

*monday	*monday	*monday	*monday	*monday	*monday	*monday	late
>	<said	<on	<	<	<	<	*monday
<	>	>	>	>	>	>	<
<on	it	<	wa	<on	report	>to	>
stock	announc	with	<on	court	releas	>and	from
trade	>that	he	at	wa	<in	meet	1
>the	>the	>the	polic	by	>the	>in	close
point	<	>that	were	hear	through	>for	yen
close	he	presid	di	he	by	begin	at
market	>that-the	bush	after	rule	<and	by	with
at	offici	hi	kill	file	publish	at	dollar
>on-the	wa	by	>and	charg	is	>the	>in
price	thei	wa	arrest	judg	am	vote	down
share	compani	govern	when	>in	edit	will	gold
wa	will	as	by	with	new	strike	up

Nation

>	>	*nation	>	*nation	>	>	>
*national	*national	>	*national	<	*national	*national	*national
secur	<	<the	<	>	<	<the	<
<	park	<of-the	bank	other	<the	associ	convent
council	forest	largest	new	european	african	<	democrat
former	<the	<in-the	<the	trade	parti	<of-the	republican
<the	at	at	republ	industri	congress	product	<the
advis	yellowston	wa	*nationally	two	assembl	gross	committe
defens	wildlif	is	holidai	>to	elect	institut	televis
drug	laboratori	most	it	12	govern	by	<of-the
<of	endow	capit	first	ar	front	<a	<a
our	anthem	aeronaut	<a	econom	<of-the	educ	*nationally
interest	<of-the	first	newspap	have	union	health	leagu
as	<a	it	radio	with	leader	percent	he
he	press	<for-the	at	western	by	center	organ

Table 5 Legend of the syntax in the table of topics

Syntax	Explanation
*⟨*word*⟩	Target word form (ex: *banking)
<⟨*stopwords*⟩	Neighboring stop words at the left of the target word (ex: <of-the)
>⟨*stopwords*⟩	Neighboring stop words at the left of the target word (ex: > means no stop word)
⟨*wordstem*⟩	Context content word (ex: insur)

Appendix 2: Treatment of WT10G topics 451–550

The following treatment was made on topic 451–550 of the TREC WT10G collection:

angioplast~~7~~y
~~do~~ beavers live in salt water
~~does~~ stress cause obesity
~~information about~~ the peer gynt suite
~~how are~~ tornadoes formed
~~how does~~ water get into the atmosphere
~~how does a~~ hygrometer measure the humidy in the atmosphere
~~how~~ e-mail bennefits businesses
~~how is~~ cancer related to cell reproduction
~~how is~~ water supplied to the mojave desert region
~~how to~~ erase scars
~~how was the~~ black plague stopped
nativity_scenes

~~whan did~~ jackie robinson appear at his first game
~~what did~~ babe ruth do in the 1920's
~~what is a~~ bengals cat
~~what is~~ the compostion of zirconium
~~where can i find~~ growth rates for the pine tree
~~where can i find information about~~ kappa alpha psi
~~where can i find information on the~~ decade of the 1920's
~~where is~~ the eldorado casino in reno

References

Bai, J., Song, D., Bruza, P., Nie, J. Y., & Cao, G. (2005). Query expansion using term relationships in language models for information retrieval. In *CIKM'05 proceedings* (pp. 688–695).

Berger, A., & Lafferty, J. (1999). Information retrieval as statistical translation. In *SIGIR'99 proceedings* (pp. 222–229).

Blei, D. M., & Lafferty, J. D. (2009). Topic models. *Text Mining: Classification, clustering, and applications* (Vol. 10, p. 71). London, England: Taylor & Francis.

Blei, D., & Lafferty, J. (2006). Correlated topic models. In Y. Weiss, B. Schölkopf, & J. Platt (Eds.), *Blei06CTM proceedings. Advances in Neural Information Processing Systems* (Vol. 18, pp. 147–154). Cambridge, MA: MIT Press.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research, 3*, 993–1022.

Brosseau-Villeneuve, B., Kando, N., & Nie, J. Y. (2011). Construction of context models for word sense disambiguation. *Information and Media Technologies, 6*(3), 701–729.

Brown, P. F., Pietra, S. A. D., Pietra, V. J. D., & Mercer, R. L. (1991). Word-sense disambiguation using statistical methods. In *ACL'91 proceedings* (pp. 264–270).

Cai, J. F., Lee, W. S., & Teh, Y. W. (2007). Nus-ml: Improving word sense disambiguation using topic features. In *SemEval'07* (pp. 249–252).

Cao, G., Nie, J. Y., Gao, J., & Robertson, S. (2008). Selecting good expansion terms for pseudo-relevance feedback. In: *SIGIR'08 proceedings* (pp. 243–250).

Croft, B., Metzler, D., & Strohman, T. (2009). *Search engines: Information retrieval in practice*. Boston: Addison-Wesley.

Croft, W., Metzler, D., & Strohmann, T. (2010). *Search engines: Information retrieval in practice*. London, UK: Pearson.

Cui, H., Wen, J. R., Nie, J. Y., & Ma, W. Y. (2002). Probabilistic query expansion using query logs. In *WWW'02 proceedings* (pp. 325–332).

Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., & Harshman, R. A. (1990). Indexing by latent semantic analysis. *Journal of the American Society of Information Science, 41*, 391–407.

Doyle, G., & Elkan, C. (2009). Accounting for burstiness in topic models. In *ICML'09 proceedings* (pp. 281–288).

Gale, W. A., Church, K. W., & Yarowsky, D. (1992). One sense per discourse. In *HLT'91 proceedings* (pp. 233–237).

Gao, J., Nie, J. Y., Wu, G., & Cao, G. (2004). Dependence language model for information retrieval. In *SIGIR'04 proceedings* (pp. 170–177).

Gaustad, T. (2001). Statistical corpus-based word sense disambiguation: Pseudowords vs. real ambiguous words. In *Companion volume to the ACL'01 proceedings* (pp. 61–66).

Gonzalo, J., Verdejo, F., Chugur, I., & Cigarrin, J. (1998). Indexing with wordnet synsets can improve text retrieval. In *COLING/ACL'98 workshop on the usage of WordNet for NLP* (pp. 38–44).

Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval, SIGIR'99* (pp. 50–57). New York, NY, USA: ACM

Ide, N., & Véronis, J. (1998). Introduction to the special issue on word sense disambiguation: The state of the art. *Computational Linguistics, 24*(1), 2–40.

Kilgarriff, A. (1997). I don't believe in word senses. *Computers and the Humanities, 31*(2), 91–113.

Kim, S. B., Seo, H. C., & Rim, H. C. (2004). Information retrieval using word senses: Root sense tagging approach. In *SIGIR'04 proceedings* (pp. 258–265).

Krovetz, R., & Croft, W. B. (1992). Lexical ambiguity and information retrieval. *ACM Transactions on Information Systems, 10*, 115–141.
Lafferty, J., & Zhai, C. (2001). Document language models, query models, and risk minimization for information retrieval. In *SIGIR'01* (pp. 111–119).
Lavrenko, V., & Croft, W. B. (2001). Relevance based language models. In *SIGIR'01 proceedings* (pp. 120–127).
Li, W., & Mccallum, A. (2006). Pachinko allocation: Dag-structured mixture models of topic correlations. In *ICML'06 proceedings* (pp. 577–584).
Lu, Y., Mei, Q., & Zhai, C. (2011). Investigating task performance of probabilistic topic models: An empirical study of plsa and lda. *Information Retrieval Journal, 14*, 178–203.
Lund, K., & Burgess, C. (1996). Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instrumentation, and Computers, 28*, 203–208.
Lv, Y., & Zhai, C. (2009). Positional language models for information retrieval. In *SIGIR'09 proceedings* (pp. 299–306).
Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to information retrieval. Cambridge: Cambridge University Press. http://nlp.stanford.edu/IR-book/.
Metzler, D., & Croft, W. B. (2004). Combining the language model and inference network approaches to retrieval. *Information Processing & Management, 40*, 735–750.
Metzler, D., & Croft, W. B. (2005). A Markov random field model for term dependencies. In *SIGIR'05 proceedings* (pp. 472–479).
Navigli, R. (2009). Word sense disambiguation: A survey. *ACM Computing Surveys, 41*(2), 1–69.
Okumura, M., Shirai, K., Komiya, K., & Yokono, H. (2010). Semeval-2010 task: Japanese wsd. In *Proceedings of the 5th international workshop on semantic evaluation* (pp. 69–74). Uppsala, Sweden: Association for Computational Linguistics.
Salton, G., Allan, J., & Buckley, C. (1993). Approaches to passage retrieval in full text information systems. In *SIGIR'93 proceedings* (pp. 49–58).
Sanderson, M. (1994). Word sense disambiguation and information retrieval. In *SIGIR'94 proceedings* (pp. 142–151).
Sanderson, M. (2000). Retrieving with good sense. *Information Retrieval, 2*, 49–69.
Sanderson, M., & Van Rijsbergen, C. J. (1999). The impact on retrieval effectiveness of skewed frequency distributions. *ACM Transactions on Information Systems, 17*, 440–465.
Schütze, H. (1998). Automatic word sense discrimination. *Computational Linguistics, 24*, 97–123.
Schutze, H., & Pedersen, J. O. (1995). Information retrieval based on word senses. In *SDAIR'95 proceedings* (pp. 161–175).
Song, F., & Croft, W. B. (1999). A general language model for information retrieval. In *CIKM'99 proceedings* (pp. 316–321).
Srikanth, M., & Srihari, R. (2002). Biterm language models for document retrieval. In *SIGIR'02 proceedings* (pp. 425–426).
Stokoe, C. (2005). Differentiating homonymy and polysemy in information retrieval. In *HLT'05 proceedings* (pp. 403–410).
Stokoe, C., Oakes, M. P., & Tait, J. (2003). Word sense disambiguation in information retrieval revisited. In *SIGIR'03 proceedings* (pp. 159–166).
Voorhees, E. M. (1993). Using wordnet to disambiguate word senses for text retrieval. In *SIGIR'93 proceedings* (pp. 171–180).
Voorhees, E. M. (2004). Overview of the trec 2004 robust retrieval track. In *TREC'04* (p. 13).
Wei, X., & Croft, W. B. (2006). LDA-based document models for ad-hoc retrieval. In *SIGIR'06 proceedings* (pp. 178–185).
Xu, J., & Croft, W. B. (1996). Query expansion using local and global document analysis. In *SIGIR'96 proceedings* (pp. 4–11).
Zhao, J., & Yun, Y. (2009). A proximity language model for information retrieval. In *SIGIR'09 proceedings* (pp. 291–298).