UNIBA-SENSE at CLEF 2008: SEmantic N-levels Search Engine

Pierpaolo Basile, Annalina Caputo and Giovanni Semeraro
Department of Computer Science - Univerisity of Bari (ITALY)
{basilepp,acaputo,semeraro}@di.uniba.it

Abstract

This paper presents evaluation experiments conducted at the University of Bari for the Ad-Hoc Robust WSD task of the Cross-Language Evaluation Forum (CLEF) 2008. The evaluation was performed using SENSE (SEmantic N-levels Search Engine) [2]. SENSE tries to overcome the limitations of the ranked keyword approach by introducing *semantic levels*, which integrate (and not simply replace) the lexical level represented by keywords.

We show how SENSE is able to manage documents indexed at two separate levels, keywords and word meanings, as well as to combine keyword search with semantic information provided by the other indexing levels, in an attempt of improving the retrieval performance.

Two types of experiments have been performed, by exploiting only one indexing level and exploiting all indexing levels at the same time. The experiments performed combining keywords and word meanings, extracted from the WordNet lexical database, show the promise of the idea and point out the value of our intuition.

In particular the results confirm our hypothesis that the combination of two indexing levels outperforms a single level. Indeed, an improvement of 35% in precision whose obtained by adopting the N-levels model with respect to the results obtained by exploiting the indexing level based only on keywords. Moreover, the Precision-Recall curve shows that the N-levels model outperforms the keywords level at all values of recall.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Indexing methods; H.3.3 Retrieval models; H.3.4 Performance evaluation (efficiency and effectiveness)

General Terms

Information Retrieval, Performance, Experimentation

Keywords

Information Retrieval, Evaluation, Cross-language Information Retrieval

1 Introduction

Information Retrieval (IR) systems are generally concerned with the selection of documents, from a fixed collection, which satisfy a user's one-off information need (query). The traditional search strategy performed by IR systems is ranked keyword search: For a given query, a list of documents,

ordered by *relevance*, is returned. Relevance computation is primarily driven by a string-matching operation: If any query word is found in a document belonging to the collection, a match is made and the document is considered as relevant.

Ranked keyword search has been quite successful in the past, in spite of its obvious limits basically due to polysemy, the presence of multiple meanings for one word, and synonymy, different words having the same meaning. The result is that, due to synonymy, relevant documents can be missed if they do not contain the exact query keywords, while, due to polysemy, wrong documents could be deemed as relevant. These problems call for alternative methods that work not only at the lexical level of the documents, but also at the *meaning* level.

Any attempt to work at the meaning level must solve the problem that, while words occur in a document, meanings do not, since they are often hidden behind words. For example, for the query "apple", some users may be interested in documents dealing with "apple" as a "fruit", while some other users may want documents related to "Apple computers". Some linguistic processing is needed in order to provide a more powerful "interpretation" both of the user needs behind the query and of the words in the document collection. This linguistic processing may result in the production of *semantic information* that provide machine readable insights into the meaning of the content.

As shown by the previous example, named entities (people, organizations, locations, etc.) mentioned in the documents constitute important part of their semantics. Therefore, in our interpretation, semantic information could be captured from a text by looking at *word meanings*, as they are described in a reference dictionary (e.g. WordNet [5]), as well as *named entities*.

We propose an IR system which manages documents indexed at multiple separate levels: keywords, senses (word meanings), and entities. The system is able to combine keyword search with semantic information provided by the two other indexing levels. In particular, for each level:

1. a *local scoring function* is used in order to weigh elements belonging to that level according to their informative power;
2. a *local similarity function* is used in order to compute document relevance by exploiting the above-mentioned scores.

Finally, a *global ranking function* is defined in order to combine document relevance computed at each level.

The rest of the paper is structured as follows. The N-levels model used in SENSE is described in Section 2, while Section 3 presents an overview of the meaning level. A brief description of the global ranking function is given in Section 4, followed by the details of the system setup for the CLEF competition in Section 5. Finally the experiments are described in Section 6. Conclusions and future work close the paper.

2 N-levels model

According to [1], an IR model is a quadruple:

$$\langle D, Q, F, R(q,d)\rangle$$

where:

- D is a set composed of logical views (or representations) for the documents in the collection;
- Q is a set composed of logical views (or representations) for user information needs. Such representations are called queries;
- F is a framework for modeling document representations, queries, and their relationships;
- $R(q,d)$ is a ranking function which associates a real number with a query $q \in Q$ and a document representation $d \in D$. Such a ranking defines an ordering among the documents with respect to the query q.

For the classic vector model, the framework F is composed of a t-dimensional vectorial space and standard linear algebra operations on vectors. In this model, tf-idf schemes are used to weigh index terms in documents D and queries Q. Term weights are used to compute the *degree of similarity* between each document d in the collection and the user query q, according to a ranking function $R(q, d)$.

We propose an extension of the classical Vector Space Model [6] called *N-levels* model in which documents are represented at different levels. Each level corresponds to a *logical view* that aims at describing one of the possible spaces in which documents are represented.

The *lexical* space of the vector model is retained and *semantic* spaces are added, each one providing different information about the meaning of the content.

Each level is described by means of a specific type of *features*, where each feature is defined as a prominent attribute or aspect of the document. The model is currently implemented in SENSE, SEmantic N-levels Search Engine, an IR system in which three different levels are considered, corresponding to as many different types of features: *keywords*, *word meanings* and *named entities*. Each document at each level is represented by a *Bag-of-Features (BOF)*, a vector of weights assigned to homogeneous features.

More formally, given $D = \{d_1, \ldots, d_{|D|}\}$ the document collection and N the number of levels, the document d_k is represented by N vectors:

$$\overrightarrow{BoF_k^i} = (w_{1,k}^i, \ldots, w_{|V_i|,k}^i) \quad i = 1, \ldots, N \tag{1}$$

where V_i is the vocabulary of the features at the i-th level, and $w_{m,k}^i$ is the weight of the m-th feature at the i-th level for document d_k.

Analogously, N query vectors (one for each level) are used for representing queries. The N query vectors are not necessarily extracted simultaneously from the original keyword query issued by the user: A query vector can be obtained when needed. For example, the ranked list of documents for the query "Apple growth" might contain documents related to both the growing of computer sales by Apple Inc. and the growth stages of apple trees. Then, when the system collects the user feedback (for instance, a click on a document in which "Apple" has been recognized as a named entity), the query vector for the named entities level will be produced.

Given these representations, we need to define a strategy to compute both $w_{m,k}^i$ and $R(q, d)$ (the degree of similarity between query and document). The weighting scheme for computing $w_{m,k}^i$ must be different for each type of feature. The adoption of a simple adjustment of tf-idf for semantic levels would result in a loss of the semantics that we want to capture by our model. More advanced strategies should be adopted in order to take into account the inherent informative power of each specific kind of feature. We call these strategies *local scoring functions*. Section 3.1 describes the local scoring function defined for weighting features (represented as WordNet synsets) at the *word meaning* level (hereafter meaning level, for brevity). As regards queries, binary weights are adopted.

We have also to extend the notion of relevance $R(q, d)$ in order to enhance keyword search with semantic information. Therefore, the degree of similarity between q and d must be evaluated at each level by defining a proper *local similarity function* that computes document relevance according to the weights defined by the corresponding local scoring function. Section 3.2 describes the local similarity function defined for the *word meaning* level. Since the final goal is to obtain a *single* list of documents ranked in decreasing order of relevance, a *global ranking function* is needed to merge all the result lists that come from each level. This function is independent of both the number of levels and the specific local scoring and similarity functions because it takes as input N ranked lists of documents and produces a unique merged list of the most relevant documents. Section 4 describes the adopted global ranking function.

To meet all the requirements of the proposed model, we implemented an extension of the Lucene API [1]. Lucene is a full-featured text search engine library that implements the vector space model.

[1] http://lucene.apache.org/

3 Meaning Level

In SENSE, features at the meaning level are *synsets* obtained from WordNet, a semantic lexicon for the English language. It groups English words into sets of synonyms called *synsets*, provides short general definitions (*glosses*), and records various semantic relations between these synonym sets. WordNet distinguishes between nouns, verbs, adjectives and adverbs because they follow different grammatical rules. Each synset is assigned with a unique identifier and contains a set of synonymous words or collocations; different senses of a word are in different synsets. The meaning of a synset is further clarified by short defining glosses. A typical example of a synset with gloss is:

01722233 ***good, right, ripe*** – *(most suitable or right for a particular purpose; "a good time to plant tomatoes"; "the right time to act"; "the time is ripe for great sociological changes")*

In order to assign synsets to words, the original system adopted a Word Sense Disambiguation (WSD) strategy. In the case of CLEF, the system used the synsets provided by the organizers of the Ad-Hoc Robust WSD task. The provided documents contain for each word a list of the possible synsets with a confidence factor. We use this factor to weigh the synset in the meaning index structure.

The idea behind the adoption of WSD is that each document is represented, at the meaning level, by the senses conveyed by the words in its content, together with their respective occurrences. Documents are represented by using a synset-based vector space. Consequently, the BOF at the meaning level is indeed a *bag-of-synsets*. The vocabulary at this level is the set of distinct synsets recognized by the WSD procedure in the collection, while $w^i_{m,k}$ in (1) is the weight of the m-th synset for document d_k, computed according to the local scoring function defined in the following section.

3.1 Synset Scoring Function

Given a document d_i and its synset representation computed by the WSD procedure, $X = [s_1, s_2, \ldots, s_k]$, the basic idea is to compute a *partial* weight for each $s_j \in X$, and then to improve this weight by finding out some relations among synsets belonging to X.

The partial weight, called SFIDF (synset frequency, inverse document frequency), is computed according to a strategy resembling the tf-idf score for words:

$$\text{SFIDF}(s_j, d_i) = \underbrace{\text{TF}(s_j, d_i)}_{\text{synset frequency}} \cdot \underbrace{log\frac{\mid C \mid}{n_j}}_{\text{IDF}} \tag{2}$$

where $\mid C \mid$ is the total number of documents in the collection and n_j is the number of documents containing the synset s_j. $\text{TF}(s_j, d_i)$ computes the frequency of s_j in document d_i.

Finally, the synset confidence factor (α) is used to weigh the SFIDF value. Thus, the final local score for synset s_j in d_i is:

$$\text{SFIDF}(s_j, d_i) \cdot (1 + \alpha) \tag{3}$$

3.2 Synset Similarity Function

The local similarity functions for both the meaning and the keyword levels are computed using a modified version of the LUCENE default document score. For the meaning level, both query and document vectors contain synsets instead of keywords. Given a query q and a document d_i, the synset similarity is computed as:

$$synsim(q, d_i) = C(q, d_i) \cdot \sum_{s_j \in q} (\text{SFIDF}(s_j, d_i)(1 + \alpha) \cdot N(d_i)) \tag{4}$$

where:

- $\text{SFIDF}(s_j, d_i)$ and α are computed as described in the previous section;

- $C(q, d_i)$ is the number of query terms in d_i.
- $N(d_i)$ is a factor that takes into account document length normalization;

4 Global Ranking

Given a query q, each local similarity function produces a local ranked list of relevant documents. All the local lists must be merged in order to give a single ranked list to the user. The global ranking function is devoted to this task.

Algorithms for merging ranked lists are widely used by meta-search engines, which send user requests to several search engines and aggregate results into a single list [4]. Our strategy for defining the *global ranking function* is thus inspired by prior work on meta-search engines.

Formally, we define:

- U: the universe, that is the set containing all the distinct documents in the local lists;
- τ_j=$\{\ x_1 \geq x_2 \geq \ldots \geq x_n\ \}$: the j-th local list, $j = 1, \ldots, N$, defined as an ordered set S of documents, $S \subseteq U$, $\geq$ is the ranking criterion defined by the j-th local similarity function;
- $\tau_j(x_i)$: a function that returns the position of x_i in the list τ_j;
- $s^{\tau_j}(x_i)$: a function that returns the score of x_i in τ_j;
- $w^{\tau_j}(x_i)$: a function that returns the weight of x_i in τ_j.

Two different strategies can be adopted to obtain $w^{\tau_j}(x_i)$, based on the score or the position of x_i in the list τ_j. Since local similarity functions may produce scores varying in different ranges, and the cardinality of lists can be different, a normalization process (of scores and positions) is necessary in order to produce weights that are comparable.

The aggregation of lists in a single one requires two steps: The first one produces the N normalized lists and the second one merges the N lists in a single one denoted by $\hat{\tau}$. The two steps are thoroughly described in [2]. After tuning experiments we choose to adopt Z-Score normalization and ComSUM respectively as score normalization and rank aggregation function. In particular the Z-Score normalization is computed using the following formula:

$$w^{\tau_j}(x_i) = \frac{s^{\tau_j}(x_i) - \mu_{s^{\tau_j}}}{\sigma_{s^{\tau_j}}}$$

Regarding ComSUM list aggregation method, the score of document x_i in the global list is computed by summing all the normalized scores for x_i:

$$\psi(x_i) = \sum_{\tau_j \in R} w^{\tau_j}(x_i)$$

where R is the set of all local lists.

5 System Setup

We adopted the SENSE model to build our IR system for CLEF evaluation. We used two different levels: keyword level using word stems and word meaning level using WordNet synsets. All the SENSE components involved in the experiments are implemented in JAVA using the last available version of Lucene API (2.3.2). Experiments were run on an Intel Core 2 Quad processor at 2.4 GHz, operating in 32 bit mode, running Linux (UBUNTU 7.10), with 2 GB of main memory.

In according to CLEF guidelines we performed two different tracks of experiments: Ad-Hoc Mono-language and Cross-language. Each track required two different evaluations: with and without synsets. We exploited several combination of levels and queries expansion methods, especially for the meaning level. All query expansion methods are automatic and do not require manual operations. Moreover we used different boosting factors for each field contained into the topic. In this way we give more importance to the terms in the fields TITLE and DESCRIPTION.

In particular for the Ad-Hoc Mono-language track we performed the following runs:

1. **MONO1TDnus2f**: the query is built using word stems in the fields TITLE and DESCRIPTION of the topics. All query terms are joined adopting the OR boolean operator. The terms in the TITLE field are boosted using a factor 2;

2. **MONO11nus2f**: similar to the previous run but in this case we add the NARRATIVE field and adopt different term boosting values: 4 for TITLE, 2 for DESCRIPTION and 1 for NARRATIVE. These boost factors are used for all the following runs;

3. **MONO12nus2f**: for this instance we adopt the Lucene Phrase Query in addition to the query expansion described in MONO11nus2f. This kind of queries are able to exploit terms proximity in the computation of relevance score. We build proximity query using the terms contained into the TITLE and DESCRIPTION fields. In detail: for TITLE we build a proximity query using all the terms into the field, while for DESCRIPTION we build a proximity query for each sentence;

4. **MONO13nus2f**: as the previous run but we adopt a different strategy to build Phrase Query. We exploit PoS-tag in order to build proximity queries. We produce a proximity query for each sequence of PoS-tags that matches the following patterns: *adjective-noun-verb*, *verb-adjective-noun*, *verb-noun*, *noun-verb* and *adjective-noun.* For example into the sentence: *'The wrapping artist Christo took two weeks...'* we build a proximity query using the following terms: *"artist Christo took"*;

5. **MONO14nus2f**: this experiment adopts a combination of all the previous methods;

6. **MONOwsd1nus2f**: the query is built expanding the synsets in the TITLE and DESCRIPTION fields of the topics. This run exploits the hypernyms and hyponyms. In particular, we include only the direct hyponyms and the hypernyms that have a path less or equal to two. For synsets we adopt a different boost factor taking into account both the field and synsets distance;

7. **MONOwsd11nus2f**: in this instance each word is expanded using the whole set of synsets in WordNet and we compute a boosting factor using the ZIPF distribution that approximates properly the natural distribution of meanings. The ZIPF formula is:

$$f(k; N; s) = \frac{1/k^s}{\sum_{n=1}^{N} 1/n^s}$$

   where:

   - N is the number of synsets;
   - k is the synset rank. The synsets in WordNet are ranked according to the their frequency in a reference corpus;
   - s is the value of exponent characterizing the distribution: after tuning experiments we set s equal to 2.

8. **MONOwsd12nus2f**: in this experiment we exploit the N-levels architecture of SENSE. For keyword level we adopt the query expansion described in MONO14nus2f and for word meaning level the MONOwsd1nus2f;

9. **MONOwsd13nus2f**: as the previous run but, for the word meaning level we adopt the method described in MONOwsd11nus2f.

For the Ad-Hoc Cross-language track we performed the following runs:

1. **CROSS1TDnus2f**: the query is built using word stems in TITLE and DESCRIPTION fields of the topics. In Cross-language track the topics are in Spanish, thus a translation of terms in English is required. The SENSE system was not developed for Cross-language

retrieval and in this instance we adopted a very trivial method in order to translate the query in English. We exploited WordNet dictionary to translate a word. In detail, we query Spanish WordNet using the Spanish word w_s and retrieve the whole set of synsets S related to the word w_s; then we use the set S to query English WordNet and retrieve, for each synset in S, the set of the English synonyms W_e. Finally, we build the query using the words in W_e. The boost factors have the same values used in the Mono-language track;

2. **CROSS1nus2f**: as described in the previous run adding the NARRATIVE field;

3. **CROSSwsd1nus2f**: in this case we adopt the same method presented in MONOwsd1nus2f but we use directly the synsets in Spanish Topic. It is important to notice that terms in a Spanish query are disambiguated using the first sense in Spanish WordNet;

4. **CROSSwsd11nus2f**: in this instance we exploit the N-levels architecture. For the keyword level we adopt the method described in CROSS1nus2f and for word meaning level the method proposed in MONOwsd1nus2f;

5. **CROSSwsd12nus2f**: this run differs from the CROSSwsd11nus2f for the use of a different Spanish-English translation method. We use directly the Spanish WordNet synset instead of the Spanish word. We query English WordNet using the synsets into the topic and retrieve, for each synset, the set of synonymous English words.

For all the runs we remove the stop words from both the index and the topics. In particular, we built a different stop words list for topics in order to remove non-informative words such as *find*, *reports*, *describe* that occur with high frequency in topics and are poorly discriminating.

6 Experimental Session

The experiments were carried out on the CLEF Ad-Hoc WSD Robust dataset derived from the English CLEF data, which comprises corpora from "Los Angeles Times" and "Glasgow Herald", amounting to 166,726 documents and 160 topics in English and Spanish. The relevance judgments were taken from CLEF.

The goal of our evaluation is to prove that the combination of two levels outperforms a single level. In particular, the combination of keyword and meaning levels is more effectiveness than the keyword level alone.

To measure retrieval performance, we adopted Mean-Average-Precision (MAP) calculated by the CLEF organizers using the DIRECT system on the basis of 1,000 retrieved items per request. Table 1 shows the results for each run with an overview on the exploited features.

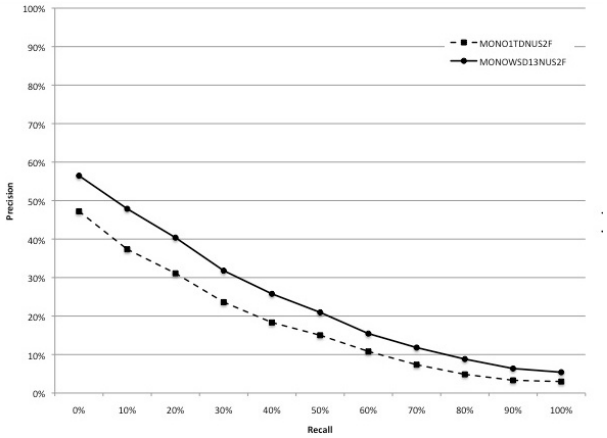

Figure 1: Precision vs. Recall graph

Run	MONO	CROSS	N-levels	WSD	*MAP*
MONO1TDnus2f	X	-	-	-	0.168
MONO11nus2f	X	-	-	-	*0.192*
MONO12nus2f	X	-	-	-	0.145
MONO13nus2f	X	-	-	-	0.154
MONO14nus2f	X	-	-	-	0.068
MONOwsd1nus2f	X	-	-	X	0.180
MONOwsd11nus2f	X	-	-	X	0.186
MONOwsd12nus2f	X	-	X	X	**0.220**
MONOwsd13nus2f	X	-	X	X	**0.227**
CROSS1TDnus2f	X	X	-	-	0.025
CROSS1nus2f	X	X	-	-	0.015
CROSSwsd1nus2f	X	X	-	X	0.071
CROSSwsd11nus2f	X	X	X	X	0.060
CROSSwsd12nus2f	X	X	X	X	**0.072**

Table 1: Results of the performed experiments

The results confirm our hypothesis: The combination of two levels outperforms a single level. In particular, the combination of keyword and meaning levels (MONOwsd12nus2f and MONOwsd13nus2f) is more effectiveness than the single keywords level (MONO1TDnus2f and MONO11nus2f). If we consider MONO1TDnus2f as baseline, we obtain an improvement of 35% in precision using the N-levels model (MONOwsd13nus2f). The behavior of the two systems is shown in Figure 1: The N-levels model outperforms the keyword level at all values of recall.

It is interesting to notice that just the use of the word meaning level alone is able to outperform the keyword level. This result has a motivation: We chose to index all the synsets for each word (not only the synset with the highest confidence factor). This intuition makes the retrieval process easier.

Regarding the Cross-language track, our system achieves a low precision. This was an expected result because our system is not designed specifically for this kind of task. Moreover, the method adopted for topic translation is based only on the use of WordNet as dictionary. In particular, performance of the Cross-language without WSD (experiments: CROSS1TDnus2f and CROSS1nus2f) are not satisfying because the system exploits only keywords and the translation process introduces a lot of wrong terms into the query, producing a noise effect. Conversely, the word meaning level is able to help the retrieval process, as shown in CROSSwsd1nus2f where we used only the word meaning level (without keywords). In the second attempt (CROSSwsd11nus2f) we combined the keyword level with the word meaning level obtaining worse results due to the keyword translation method (as in CROSS1TDnus2f). Finally, we tried to translate the Spanish words using directly the synsets obtaining a good result with respect to the previous one.

We noticed that our system has a low precision with respect to the other participants to the CLEF competition. This is due to the standard relevance function implemented in Lucene and this result was expected. In particular, Lucene performance decreases when the number of terms in the query grows. In fact, the experiment MONO14nus2f produces large queries and results point out that the system achieves a low precision in this experiment with respect to the others that rely exclusively on keywords. This problem also affects the Cross-language experiments because we translate a Spanish word using all the possible English translations (CROSS1TDnus2f) producing a query with a lot of terms. Details concerning this well known behavior of Lucene can be found in [3]. Nonetheless, the goal of our evaluation was to prove the effectiveness of the N-levels model and the experiments confirm our hypothesis.

7 Conclusion and Future Work

We have described and tested SENSE (SEmantic N-levels Search Engine), a semantic N-levels IR system which manages documents indexed at multiple separate levels: keywords and meanings. The system is able to combine keyword search with semantic information provided by the other indexing level.

The distinctive feature of the system is that, differently from the previous approaches, an adaptation of the vector space model is proposed to integrate, rather than simply replace, the lexical space with semantic spaces. We provided a detailed description of the SENSE model, by defining a local scoring function, a local similarity function for synsets and a global ranking function in order to merge rankings produced by different levels.

We performed an intensive evaluation using the CLEF Ad-Hoc Robust WSD dataset. This dataset supplies both words and synsets for each document and it is the ideal framework to evaluate the N-levels architecture. The experiments show that the N-levels model is effective when the word meaning level is involved.

As future research we plan to improve the performance of the system. We can achieve this goal adopting two different strategies: The former involves the change of the relevance function implemented in Lucene; the latter exploits the possibility to replace vector space model with a more effective IR model.

References

[1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, 1999.

[2] Pierpaolo Basile, Annalina Caputo, Anna Lisa Gentile, Marco Degemmis, Pasquale Lops, and Giovanni Semeraro. Enhancing semantic search using n-levels document representation. In Stephan Bloehdorn, Marko Grobelnik, Peter Mika, and Duc Thanh Tran, editors, *SemSearch*, volume 334 of *CEUR Workshop Proceedings*, pages 29–43. CEUR-WS.org, 2008.

[3] D. Cohen, E. Amitay, and D. Carmel. Lucene and juru at trec 2007: 1-million queries track. In *Proceedings of the 16th Text REtrieval Conference (TREC 2007)*, November 2007.

[4] Mohamed Farah and Daniel Vanderpooten. An outranking approach for rank aggregation in information retrieval. In Wessel Kraaij, Arjen P. de Vries, Charles L. A. Clarke, Norbert Fuhr, and Noriko Kando, editors, *SIGIR*, pages 591–598. ACM, 2007.

[5] G. A. Miller. Wordnet: a lexical database for english. *Commun. ACM*, 38(11):39–41, 1995.

[6] G. Salton. *The SMART Retrieval System - Experiments in Automatic Document Processing*. Prentice Hall, Englewood Cliffs, NJ, 1971.