Learning to Identify Definitions using Syntactic Features

Ismail Fahmi and **Gosse Bouma**
Information Science
Groningen University
{i.fahmi,g.bouma}@rug.nl

Abstract

This paper describes an approach to learning concept definitions which operates on fully parsed text. A subcorpus of the Dutch version of Wikipedia was searched for sentences which have the syntactic properties of definitions. Next, we experimented with various text classification techniques to distinguish actual definitions from other sentences. A maximum entropy classifier which incorporates features referring to the position of the sentence in the document as well as various syntactic features, gives the best results.

1 Introduction

Answering definition questions is a challenge for question answering systems. Much work in QA has focused on answering factoid questions, which are characterized by the fact that given the question, one can typically make strong predictions about the type of expected answer (i.e. a date, name of a person, amount, etc.). Definition questions require a different approach, as a definition can be a phrase or sentence for which only very global characteristics hold.

In the CLEF 2005 QA task, 60 out of 200 questions were asking for the definition of a named entity (a person or organization) such as *Who is Goodwill Zwelithini?* or *What is IKEA?* Answers are phrases such as *current king of the Zulu nation*, or *Swedish home furnishings retailer*. For answering definition questions restricted to named entities, it generally suffices to search for noun phrases consisting of the named entity and a preceding or following nominal phrase. Bouma et al. (2005) extract all such noun phrases from the Dutch CLEF corpus off-line, and return the most frequent heads of co-occurring nominal phrases expanded with adjectival or prepositional modifiers as answer to named entity definition questions. The resulting system answers 50% of the CLEF 2005 definition questions correctly.

For a Dutch medical QA system, which is being developed as part of the IMIX project[1], several sets of test questions were collected. Approximately 15% of the questions are definition questions, such as *What is a runner's knee?* and *What is cerebrovascular accident?*. Answers to such questions (asking for the definition of a concept) are typically found in sentences such as *A runner's knee is a degenerative condition of the cartilage surface of the back of the knee cap, or patella* or *A cerebrovascular accident is a decrease in the number of circulating white blood cells (leukocytes) in the blood.* One approach to finding answers to concept definitions simply searches the corpus for sentences consisting of a subject, a copular verb, and a predicative phrase. If the concept matches the subject, the predicative phrase can be returned as answer. A preliminary evaluation of this technique in Tjong Kim Sang et al. (2005) revealed that only 18% of the extracted sentences (from a corpus consisting of a mixture of encyclopedic texts and web documents) is actually a definition. For instance, sentences such as *RSI is a major problem in the Netherlands*, *every suicide attempt is an emergency* or *an infection of the lungs is the most serious complication* are of the relevant syntactic form, but do not constitute definitions.

In this paper, we concentrate on a method for improving the precision of recognizing definition sentences. In particular, we investigate to what

[1] www.let.rug.nl/~gosse/Imix

extent machine learning techniques can be used to distinguish definitions from non-definitions in a corpus of sentences containing a subject, copular verb, and predicative phrase. A manually annotated subsection of the corpus was divided into definition and non-definition sentences. Next, we trained various classifiers using unigram and bigram features, and various syntactic features. The best classifier achieves a 60% error reduction compared to our baseline system.

2 Previous work

Work on identifying definitions from free text initially relied on manually crafted patterns without applying any machine learning technique. Klavans and Muresan (2000) set up a pattern extractor for their Definder system using a tagger and a finite state grammar. Joho and Sanderson (2000) retrieve descriptive phrases (*dp*) of query nouns (*qn*) from text to answer definition questions like *Who is qn?* Patterns such as '*dp* especially *qn*', as utilized by Hearst (1992), are used to extract names and their descriptions.

Similar patterns are also applied by Liu et al. (2003) to mine definitions of topic-specific concepts on the Web. As an additional assumption, specific documents dedicated to the concepts can be identified if they have particular HTML and hyperlink structures.

Hildebrandt et al. (2004) exploit surface patterns to extract as many relevant "nuggets" of information of a concept as possible. Similar to our work, a copular pattern NP_1 *be* NP_2 is used as one of the extraction patterns. Nuggets which do not begin with a determiner are discarded to filter out spurious nuggets (e.g., progressive tense). Nuggets extracted from every article in a corpus are then stored in a relational database. In the end, answering definition questions becomes as simple as looking up relevant terms from the database. This strategy is similar to our approach for answering definition questions.

The use of machine learning techniques can be found in Miliaraki and Androutsopoulos (2004) and Androutsopoulos and Galanis (2005) They use similar patterns as (Joho and Sanderson, 2000) to construct training attributes. Sager and L'Homme (1994) note that the definition of a term should at least always contain *genus* (term's category) and *species* (term's properties). Blair-Goldensohn et al. (2004) uses machine learning and manually crafted lexico-syntactic patterns to match sentences containing both a genus and species phrase for a given term.

There is an intuition that most of definition sentences are located at the beginning of documents. This lead to the use of sentence number as a good indicator of potential definition sentences. Joho and Sanderson (2000) use the position of the sentences as one of their ranking criteria, while Miliaraki and Androutsopoulos (2004), Androutsopoulos and Galanis (2005) and Blair-Goldensohn et al. (2004) apply it as one of their learning attributes.

3 Syntactic properties of potential definition sentences

To answer medical definition sentences, we used the medical pages of Dutch Wikipedia[2] as source. Medical pages were selected by selecting all pages mentioned on the *Healthcare* index page, and recursively including pages mentioned on retrieved pages as well.

The corpus was parsed syntactically by Alpino, a robust wide-coverage parser for Dutch (Malouf and van Noord, 2004). The result of parsing (illustrated in Figure 1) is a dependency graph. The Alpino-parser comes with an integrated named entity classifier which assigns distinct part-of-speech tags to person, organization, and geographical named entities.

Potential definition sentences are sentences containing a form of the verb *zijn*[3] (*to be*) with a subject and nominal predicative phrase as sisters. The syntactic pattern does not match sentences in which *zijn* is used as a possessive pronoun (*his*) and sentences where a form of *zijn* is used as an auxiliary. In the latter case, no predicative phrase complement will be found. On the other hand, we do include sentences in which the predicative phrase precedes the subject, as in *Onderdeel van de testis is de Leydig-cel* (*the Leydig cel is part of the testis*). As word order in Dutch is less strict than in English, it becomes relevant to include such non-canonical word orders as well.

A number of non-definition sentences that will be extracted using this method can be filtered by simple lexical methods. For instance, if the subject is headed by (the Dutch equivalents of) *cause, con-*

[2] `nl.wikipedia.org`

[3] Note that the example uses *ben* (the first person singular form of the verb) as root for *zijn*.

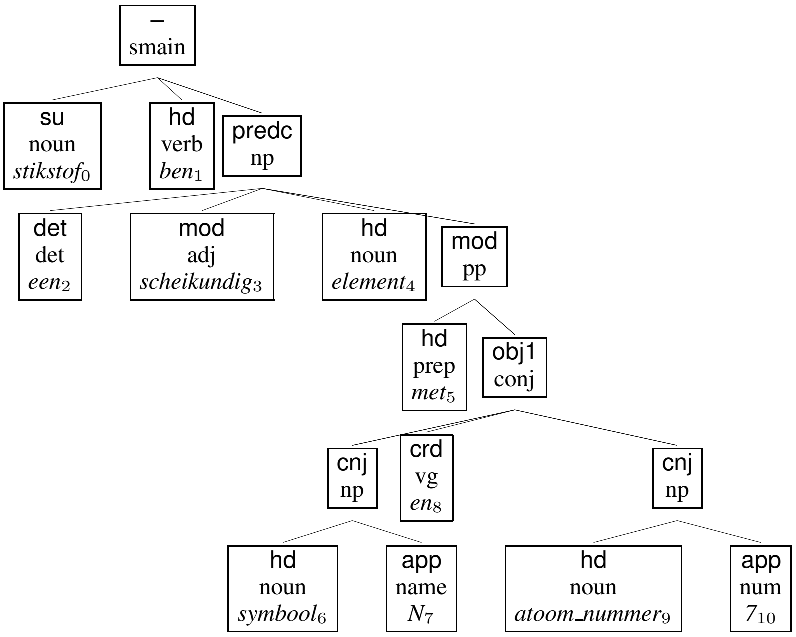

Figure 1: Parse of (the Dutch equivalent of) *Nitrogen is a chemical element with symbol N and atomic number 7.* Nodes are labelled with depedency relations and categories or part-of-speech tags, root forms, and string positions.

sequence, example, problem, result, feature, possibility, symptom, sign, etc., or contains the determiner *geen* (*no*), the sentence will not be included in the list of potential definitions.

However, even after applying the lexical filter, not all extracted sentences are definitions. In the next sections, we describe experiments aimed at increasing the accuracy of the extraction method.

4 Annotating training examples

To create evaluation and training data, 2500 extracted sentences were manually annotated as definition, non-definition, or undecided. One of the criteria for undecided sentences is that it mentions a characteristic of a definition but is not really a (complete) definition, for example, *Benzeen is carcinogeen* (*Benzene is a carcinogen*). The result of this annotation is given in Table 1. The annotated data was used both to evaluate the accuracy of the syntactic extraction method, and to training and evaluate material for the machine learning experiments as discussed in the next sections.

After discarding the undecided sentences, we are left with 2299 sentences, 1366 of which are definitions. This means that the accuracy of the extraction method using only syntax was 59%.[4]

If we take sentence postion into account as well, and classify all first sentences as definitions and all other sentences as non-definitions, a baseline accuracy of 75,9% is obtained.

It is obvious from Table 1 that the first sentences of Wikipedia lemmas that match the syntactic pattern are almost always definitions. It seems that e.g. Google's[5] *define* query feature, when restricted to Dutch at least, relies heavily on this fact to answer definition queries. However it is also obvious that definition sentences can also be found in other positions. For documents from other sources, which are not as structured as Wikipedia, the first position sentence is likely to be an even weaker predictor of definition vs. non-definition sentences.

5 Attributes of definition sentences

We aim at finding the best attributes for classifying definition sentences. We experimented with combinations of the following attributes:

Text properties: bag-of-words, bigrams, and root forms. Punctuation is included as Klavans and Muresan (2000) observe that it can be used to recognize definitions (i.e. definitions tend to con-

[4] This is considerably higher than the estimated accuracy of 18% reported in Tjong Kim Sang et al. (2005). This is probably partly due to the fact that the current corpus consists of encyclopedic material only, whereas the corpus used in Tjong Kim Sang et al. (2005) contained web material from various sources, such as patient discussion groups, as well. The latter tends to contain more subjective and context-dependent material.

[5] `google.com`

Sentence position	Def	Non-def	Undecided
first	831	18	31
other	535	915	170
Total	1366	933	201

Table 1: Number of sentences in the first and other position of documents annotated as definition, non-definition, and undecided.

tain parentheses more often than non-definitions). No stopword filtering is applied as in our experiments it consistently decreased accuracy. Note that we include all bigrams in a sentence as feature. A different use of n-grams has been explored by Androutsopoulos and Galanis (2005) who add only n-grams ($n \in \{1,2,3\}$) occurring frequently either directly before or after a target term.

Document property: the position of each sentence in the document. This attribute has been frequently used in previous work and is motivated by the observation that definitions are likely to be located in the beginning of a document.

Syntactic properties: position of each subject in the sentence (*initial*, e.g. *X is Y*; or *non-initial*, e.g. *Y is X*), and of each subject and predicative complement: type of determiner (definite, indefinite, other). These attributes have not been investigated in previous work. In our experiments, sentence-initial subjects appear in 92% of the definition sentences and and 76% of the non-definition sentences. These values show that a definition sentence with a copular pattern tends to put its subject in the beginning. Two other attributes are used to encode the type of determiner of the subject and predicative compelement. As shown in Table 2, the majority of subjects in definition sentences have no determiner (62%), e.g. *Paracetamol is een pijnstillend en koortsverlagend middel* (*Paracetamol is an pain alleviating and a fever reducing medicine*), while in non-definition sentences subject determiners tend to be definite (50%), e.g. *De werkzame stof is acetylsalicylzuur* (*The operative substance is acetylsalicylacid*). Predicative complements, as shown in Table 3, tend to contain indefinite determiners in definition sentences (64%), e.g. *een pijnstillend ... medicijn* (*a pain alleviating... medicine*), while in non-definition the determiner tends to be definite (33%), e.g. *Een fenomeen is de Landsgemeinde* (*A phenomenon is the Landsgemeinde*).

Type	Definition	Non-def
definite	23	**50**
indefinite	13	12
nodeterminer	**62**	29
other	2	9

Table 2: Percentage of determiner types of subjects in definition and non-definition sentences.

Type	Definition	Non-def
definite	23	**33**
indefinite	**64**	29
nodeterminer	8	1
other	4	28

Table 3: Percentage of determiner types of predicative complements in definition and non-definition sentences.

Named entity tags: named entity class (NEC) of subjects, e.g. location, person, organization, or no-class. A significant difference in the distribution of this feature between definition and non-definition sentences can be observed in Table 4. More definition sentences have named entity classes contained in their subjects (40.63%) compared to non-definition sentences (11.58%). We also experimented with named entity classes contained in predicative complements but it turned out that very few predicates contained named entities, and thus no significant differences in distribution between definition and non-definition sentences could be observed.

Features for lexical patterns, as used in (Androutsopoulos and Galanis, 2005), e.g. *qn which (is|was|are|were) dp*, are not added because in this experiment we investigate only a copular pattern. WordNet-based attributes are also excluded, given that coverage for Dutch (using EuroWordNet) tends to be less good than for English, and even for English their contribution is sometimes insignificant (Miliaraki and Androutsopoulos, 2004).

Type	Definition	Non-def
no-nec	59	88
location	10	4
organization	8	3
person	22	4

Table 4: Percentage of named-entity classes of subjects in definition and non-definition sentences.

word bigrams only	bigram + synt + pos
is a	first_sent
a	other_sent
are	is a
is	indef_pred
) is	no_det_subj
the	init_subj
is DIGITS	a
are the	are
this	is
or	other_det_pred
is of	) is
this/these	noninit_subj
atomic_number	def_subj
atomic_number DIGITS	the
with symbol	is DIGITS
and atomic_number	are the
that	this
chemical	or
a chemical	other_det_subj
chemical element	is of

Table 5: 20 most informative features for the systems using word bigrams only and word bigrams in combination with syntactic and sentence position features (word features have been translated into English).

We use the text classification tool Rainbow[6] (McCallum, 2000) to perform most of our experiments. Each sentence is represented as a string of words, possibly followed by bigrams, root forms, (combinations of) syntactic features, etc.

All experiments were performed by selecting only the 2000 highest ranked features according to information gain. In the experiments which include syntactic features, the most informative features tend to contain a fair number of syntactic features. This is illustrated for the configuration using bigrams, sentence position, and syntax in table 5. It supports our intuition that the position of subjects and the type of determiner of subjects and predicative complements are clues to recognizing definition sentences.

To investigate the effect of each attribute, we set up several configurations of training examples as described in Table 6. We start with using only bag-of-words or bigrams, and then combine them with other attribute sets.

Cfg	**Description**
1	using only bag-of-words
2	using only bigrams
3	combining bigrams & bag-of-words
4	adding syntactic properties to config. 3
5	adding syntactic properties & NEC to config. 3
6	adding sentence position to config. 3
7	adding root forms to config. 3
8	adding syntactic properties & sentence position to config. 3
9	adding syntactic properties, sentence position & NEC to config. 3
10	adding syntactic properties, sentence position & root forms to config. 3)
11	using all attributes (adding NEC to configuration 10)

Table 6: The description of the attribute configurations.

6 Learning-based methods

We apply three supervised learning methods to each of the attribute configurations in Table 6, namely naive Bayes, maximum entropy, and support vector machines (SVMs). Naive Bayes is a fast and easy to use classifier based on the probabilistic model of text and has often been used in text classification tasks as a baseline. Maximum entropy is a general estimation technique that has been used in many fields such as information retrieval and machine learning. Some experiments in text classification show that maximum entropy often outperforms naive Bayes, e.g. on two of three data sets in Nigam et al. (1999). SVMs are a new learning method but have been reported by Joachims (1998) to be well suited for learning in text classification.

We experiment with three kernel types of SVMs: linear, polynomial, and radial base function (RBF). Rainbow (McCallum, 2000) is used to examine these learning methods, except the RBF kernel for which libsvm (Chang and Lin, 2001) is used. Miliaraki and Androutsopoulos (2004) use a SVM with simple inner product (polynomial of first degree) kernel because higher degree polynomial kernels were reported as giving no improvement. However we want to experiment with

[6] www.cs.cmu.edu/˜mccallum/bow/rainbow/

Cfg	NB	ME	svm1[a]	svm2[b]	svm3[c]
1	85.75 ± 0.57	85.35 ± 0.77	77.65 ± 0.87	78.39 ± 0.67	81.95 ± 0.82
2	87.77 ± 0.51	88.65 ± 0.54	84.02 ± 0.47	84.26 ± 0.52	85.38 ± 0.77
3	89.82 ± 0.53	88.82 ± 0.66	83.93 ± 0.57	84.24 ± 0.54	87.04 ± 0.95
4	85.22 ± 0.35	89.08 ± 0.50	84.93 ± 0.57	85.57 ± 0.53	87.77 ± 0.89
5	85.44 ± 0.45	91.38 ± 0.42	86.90 ± 0.48	86.90 ± 0.53	87.60 ± 0.87
6	**90.26** ± **0.71**	90.70 ± 0.48	85.26 ± 0.56	86.05 ± 0.64	88.52 ± 0.92
7	88.60 ± 0.81	88.99 ± 0.51	83.38 ± 0.38	84.69 ± 0.43	87.08 ± 0.87
8	86.40 ± 0.51	**92.21** ± **0.27**	86.57 ± 0.42	87.29 ± 0.47	88.77 ± 0.77
9	87.12 ± 0.52	90.83 ± 0.43	87.21 ± 0.42	**87.99** ± **0.53**	89.04 ± 0.67
10	87.60 ± 0.38	91.16 ± 0.43	86.68 ± 0.40	86.97 ± 0.41	88.91 ± 0.68
11	86.72 ± 0.46	91.16 ± 0.35	**87.47** ± **0.40**	87.05 ± 0.63	**89.47** ± **0.67**

[a] SVM with linear kernel (Rainbow)
[b] SVM with polynomial kernel (Rainbow)
[c] SVM with RBF kernel (libsvm)

Table 7: Accuracy and standard error (%) estimates for the dataset using naive Bayes (NB), maximum entropy (ME), and three SVM settings at the different attribute configurations.

the RBF (gaussian) kernel by selecting model parameters C (penalty for misclassification) and γ (function of the deviation of the Gaussian Kernel) so that the classifier can accurately predict testing data. This experiment is based on the argument that if a complete model selection using the gaussian kernel has been conducted, there is no need to consider linear SVM, because the RBF kernel with certain parameters (C, γ) has the same performance as the linear kernel with a penalty parameter $\tilde{C}$ (Keerthi and Lin, 2003).

Given the finite dataset, we use k-fold cross-validation (k = 20) to estimate the future performance of each classifier induced by its learning method and dataset. This estimation method introduces lower bias compared to a bootstrap method which has extremely large bias on some problems (Kohavi, 1995).

7 Evaluation

We evaluated each configuration of Section 5 and each learning method of Section 6 on the dataset which consists of 1336 definitions and 963 non-definitions sentences. Table 7 reports the accuracy and standard error estimated from this experiment.

In all experiment runs, all of the classifiers in all configurations outperform our baseline (75.9%). The best accuracy of each classifier (bold) is between 11.57% to 16.31% above the baseline.

The bigram only attributes (config. 2) clearly outperform the simplest setting (bag-of-word only attributes) for all classifiers. The combination of both attributes (config. 3) achieves some improvement between 0.17% to 4.41% from configuration 2. It is surprising that naive Bayes shows the best and relatively high accuracy in this base configuration (89.82%) and even outperforms all other settings.

Adding syntactic properties (config. 4) or position of sentences in documents (config. 6) to the base configuration clearly gives some improvement (in 4 and 5 classifiers respectively for each configuration). But, adding root forms (config. 7) does not significantly contribute to an improvement. These results show that in general, syntactic properties can improve the performance of most classifiers. The results also support the intuition that the position of sentences in documents plays important role in identifying definition sentences. Moreover, this intuition is also supported by the result that the best performance of naive Bayes is achieved at configuration 6 (90.26%). Compared to the syntactic features, sentence positions give better accuracy in all classifiers.

The above results demonstrate an interesting finding that a simple attribute set which consists of bag-of-words, bigrams, and sentence position under a fast and simple classifier (e.g. naive Bayes) could give a relatively high accuracy. One explanation that we can think of is that candidate sentences have been syntactically very well extracted with our filter. Thus, the sentences are biased by the filter from which important words and bigrams of definitions can be found in most of the sentences. For example, the word and bigrams *is een (is a), een (a), zijn (are), is (is), zijn de (are the),* and *is van (is of)* are good clues to definitions and consequently have high information gain. We have to test this result in a future work on candidate definition sentences which are extracted by filters using various other syntactic patterns.

More improvement is shown when both syntactic properties and sentence position are added together (config. 8). All of the classifiers in this configuration obtain more error reduction compared to the base configuration. Moreover, the best accuracy of this experiment is shown by maximum entropy at this configuration (92.21%). This may be a sign that our proposed syntactic properties are good indicators to identify definition sentences.

Other interesting findings can be found in the addition of named entity classes to configuration 3 (config. 5), to configuration 8 (config. 9) and to configuration 10 (config. 11). In these configurations, adding NEC increases accuracies of almost all classifiers. On the other hand, adding root forms to configuration 3 (config. 7) and to configuration 8 (config. 10) does not improve accuracies. However, the best accuracies of naive Bayes (90.26%) and maximum entropy (92.21%) are achieved when named entity and root forms are not included as attributes.

We now evaluate the classifiers. It is clear from the table that SVM1 and SVM2 settings can not achieve better accuracy compared to the naive Bayes setting, while SVM3 setting marginally outperforms naive Bayes (on 6 out of 11 configurations). This result is contrary to the superiority of SVMs in many text classification tasks. Huang et al. (2003) reported that both classifiers show similar predictive accuracy and AUC (area under the ROC (Receiver Operating Characteristics) curve) scores. This performance of naive Bayes supports the motivation behind its renaisance in machine learning (Lewis, 1998).

From the three SVM settings, SVM with RBF kernel appears as the best classifier for our task in which it outperforms other SVMs settings in all configurations. This result supports the above mentioned argument that if the best C and γ can be selected, we do not need to consider linear SVM (e.g. the svm1 setting).

Among all of the classifiers, maximum entropy shows the best accuracy. It wins at 9 out of 11 configurations in all experiments. This result confirms previous reports e.g. in Nigam et al. (1999) that maximum entropy performs better than naive Bayes in some text classification tasks.

8 Conclusions and future work

We have presented an experiment in identifying definition sentences using syntactic properties and learning-based methods. Our method is concentrated on improving the precision of recognizing definition sentences. The first step is extracting candidate definition sentences from a fully parsed text using syntactic properties of definitions. To distinguish definition from non-definition sentences, we investigated several machine learning methods, namely naive Bayes, maximum entropy, and SVMs. We also experimented with several attribute configurations. In this selection, we combine text properties, document properties, and syntactic properties of the sentences. We have shown that adding syntactic properties, in particular the position of subjects in the sentence, type of determiner of each subject and predicative complement, improves the accuracy of most machine learning techniques, and leads to the most accurate result overall.

Our method has been evaluated on a subset of manually annotated data from Wikipedia. The combination of highly structured text material and a syntactic filter leads to a relatively high initial baseline.

Our results on the performance of SVMs do not confirm the superiority of this learning method for (text) classification tasks. Naive Bayes, which is well known from its simplicity, appears to give reasonably high accuracy. Moreover, it achieves a high accuracy on simple attribute configuration sets (containing no syntactic properties). In general, our method will give the best result if all properties except named entity classes and root forms are used as attributes and maximum entropy is applied as a classifier.

We are currently working on using more syntactic patterns to extract candidate definition sentences. This will increase the number of definition sentences that we can identify from text.

References

I. Androutsopoulos and D. Galanis. 2005. A practically unsupervised learning method to identify single-snippet answers to definition questions on the web. In *Human Language Technology Conference* *and Conference on Empirical Methods in Natural Language Processing (HLT-EMNLP 2005)*, Vancouver, Canada.

S. Blair-Goldensohn, K. McKeown, and A.H. Schlaikjer. 2004. Answering definitional questions: A hybrid approach. In *New Directions in Question Answering*, pages 47–58.

Gosse Bouma, Jori Mur, Gertjan van Noord, Lonneke van der Plas, and Jörg Tiedemann. 2005. Question answering for Dutch using dependency relations. In *Working Notes for the CLEF 2005 Workshop*, Vienna.

Chih-Chung Chang and Chih-Jen Lin, 2001. *LIBSVM: a library for support vector machines*. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proc. of the 14th COLING*, pages 539–545, Nantes, France.

W. Hildebrandt, B. Katz, and J.J. Lin. 2004. Answering definition questions with multiple knowledge sources. In *HLT-NAACL*, pages 49–56.

Jin Huang, Jingjing Lu, and Charles X. Ling. 2003. Comparing naive bayes, decision trees, and svm with auc and accuracy. In *ICDM '03: Proceedings of the Third IEEE International Conference on Data Mining*, Washington, DC, USA. IEEE Computer Society.

Thorsten Joachims. 1998. Text categorization with support vector machines: learning with many relevant features. In Claire N'edellec and C'eline Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, pages 137–142, Chemnitz, DE. Springer Verlag, Heidelberg, DE.

H. Joho and M. Sanderson. 2000. Retrieving descriptive phrases from large amounts of free text. In *CIKM*, pages 180–186.

S. Sathiya Keerthi and Chih-Jen Lin. 2003. Asymptotic behaviors of support vector machines with gaussian kernel. *Neural Comput.*, 15(7):1667–1689.

J.L. Klavans and S. Muresan. 2000. Definder: Rule-based methods for the extraction of medical terminology and their associated definitions from on-line text. In *American Medical Informatics Assoc 2000.*

Ron Kohavi. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *IJCAI*, pages 1137–1145.

David D. Lewis. 1998. Naive (Bayes) at forty: The independence assumption in information retrieval. In Claire Nédellec and Céline Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, pages 4–15, Chemnitz, DE. Springer Verlag, Heidelberg, DE.

B. Liu, C.W. Chin, and H.T. Ng. 2003. Mining topic-specific concepts and definitions on the web. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 251–260, New York, NY, USA. ACM Press.

Robert Malouf and Gertjan van Noord. 2004. Wide coverage parsing with stochastic attribute value grammars. In *IJCNLP-04 Workshop Beyond Shallow Analyses - Formalisms and statistical modeling for deep analyses*, Hainan.

A McCallum. 2000. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. `http://www.cs.cmu.edu/~mccallum/bow`.

S. Miliaraki and I. Androutsopoulos. 2004. Learning to identify single-snippet answers to definition questions. In *20th International Conference on Computational Linguistics (COLING 2004)*, pages 1360–1366, Geneva, Switzerland. COLING 2004.

K. Nigam, J. Lafferty, and A. McCallum. 1999. Using maximum entropy for text classification. In *IJCAI-99 Workshop on Machine Learning for Information Filtering*, pages 61–67.

Juan C. Sager and M.C. L'Homme. 1994. A model for definition of concepts. *Terminology*, pages 351–374.

Erik Tjong Kim Sang, Gosse Bouma, and Maarten de Rijke. 2005. Developing offline strategies for answering medical questions. In Diego Mollá and José Luis Vicedo, editors, *AAAI 2005 workshop on Question Answering in Restricted Domains*.