Using machine learning to perform automatic term recognition

Jody Foo, Magnus Merkel

Department of Computer and Information Science,
Linköping university
jodfo@ida.liu.se, magme@ida.liu.se

Abstract

In this paper a machine learning approach is applied to Automatic Term Recognition (ATR). Similar approaches have been successfully used in Automatic Keyword Extraction (AKE). Using a dataset consisting of Swedish patent texts and validated terms belonging to these texts, unigrams and bigrams are extracted and annotated with linguistic and statistical feature values. Experiments using a varying ratio between positive and negative examples in the training data are conducted using the annotated n-grams. The results indicate that a machine learning approach is viable for ATR. Furthermore, a machine learning approach for bilingual ATR is discussed. Preliminary analysis however indicate that some modifications have to be made to apply the monolingual machine learning approach to a bilingual context.

1. Background

Term extraction, or more specifically, Automatic Term Recognition (ATR) is a field in language technology that involves "extraction of technical terms from domain-specific language corpora." (Zhang et al., 2008). One area of application of ATR systems is the field of terminology where one task is to define concepts and decide which terms should be attached to them. Another area of application is creating domain specific dictionaries for use in for example machine translation (Merkel et al., 2009).

Closely related fields to ATR are Automatic Keyword/Keyphrase extraction, (AKE) (Turney, 2000; Hulth, 2004), and Automatic Index Generation (AIG) and Information retrieval (IR) in general. Common to these fields are the following components.

- text analysis
- selecting/filtering term candidates
- ranking term candidates

The first component involves the process of analyzing the source text(s) and attaching feature value pairs to each token in the text. The second component involves the process of selecting single or multi-word-units from a text. The selection process may be iterative, e.g. it may consist of selection and filtering in several iterations. In most cases, filtering relies on the feature-value pairs previously attached. The third component involves using one or several metrics to rank term candidates.

2. Current research

In order to decide what should be considered a term, it is necessary to analyze and attach some kind of information to the objects of consideration. In general, the kind of information produced by a method of analysis falls into one of two categories. Either it is 1) *statistical/distributional* or 2) *linguistic*. Statistical information is based on statistical analysis, e.g. word counts, probabilities, mutual information etc. Distributional properties can also be grouped into this category, e.g. corpora frequency comparisons. Linguistic information is based on linguistic analysis e.g. part of speech, semantics etc.

Although research in the fields of ATR, AKE and IR have common components, it is interesting to note that there seems to be little interaction between researchers within these fields, especially between ATR and AKE. It is also interesting to note that early ATR research (Bourigault, 1992; Ananiadou, 1994) relied on linguistic analysis, where as early work in Information retrieval relied on statistical measures (Salton and McGill, 1986). Current ATR research has however moved towards using statistical measures (Zhang et al., 2008) and current AKE research has moved towards using more linguistic analysis (Hulth, 2004). These movements, though moving towards the same center of conversion, seem to be independent of each other.

2.1. Statistical analysis

The most basic statistical measure is the frequency count, which has can be refined by normalizing, and more commonly, by adding a distributional component as with the *tf-idf* measure (Salton and McGill, 1986). In ATR a distinction between "unithood" and "termness". Unithood measures are used to determine collocation strength between units when dealing with terms that consist of more than one word. Termness measures indicate how associated to the domain a term is. Typical unithood measures are mutual information (Daille, 1994) and log-likelihood (Cohen, 1995a). Statistical measures such as the C-value/NC-value (Frantzi et al., 1998) integrate termness unithood. The C-value part of this metric is tailored to recognize termness. The NC-value takes context into account and thereby improves multiword unit recognition.

2.2. Linguistic analysis

Early term recognition systems such as (Bourigault, 1992; Ananiadou, 1994) used part-of-speech and chunking to facilitate the use part-of-speech patterns to recognize term candidates. Frameworks relying solely on linguistic analysis use hand-crafted rules such as {`DET A N`} to recognize term candidates.

2.3. Term candidate selection

The term candidate selection process is the process of selecting which of the extracted terms should be passed on to e.g. a domain expert for validation. To do this, the initial set of possible term candidates is truncated into a smaller set using a metric that measures termness. The better the termness value, the more likely it is that the extracted term candidate will be accepted by a domain-expert.

2.4. Machine learning

To the authors' knowledge, machine learning has not been applied to ATR. However, it is a common approach within AKE (Turney, 2000; Hulth, 2003; Hulth, 2004). (Turney, 2000) performed experiments using C4.5 (Quinlan, 1993) among others. (Hulth, 2004) used a rule induction system called Compumine[1]. Turney (2000) used 10 features in his C4.5 experiments, most of them linguistically uninformed, such as number of words in the phrase, frequency of the phrase and relative length of phrase. Three of the features used heuristics to determine whether the phrase was a proper noun, ended by an adjective or contained a common verb. The experiments carried out by (Hulth, 2003) used the same features as (Frank et al., 1999) but also add a string containing part-of-speech tags.

3. Problem specification

As machine learning has been successfully applied to AKE, it is relevant to examine its efficiency when applied to ATR. Though the extraction processes of ATR and AKE are similar, the two tasks are different when looking at the expected output. The task of AKE is to output a relatively short list of keywords/keyphrases that describe a document. This size of this list is between 5 and 15 keywords/keyphrases long. In ATR there is no limit on how many terms are extracted.

One problem when applying machine learning techniques is that the composition of the training data can have a drastic effect on precision and recall. Given that one property of terms in documents is that they make up only a small percentage of the total amount of tokens, it is important to examine how this property effects training results. Performing experiments with different positive/negative example ratios will add two important pieces of knowledge for future research.

1. Is a machine learning approach feasible for ATR?
2. What should the ratio be between positive and negative examples in the training data for future experiments?

In the remaining part of this paper we will describe such experiments, their results and possible directions for future research.

4. Method

The method examined in this paper uses Ripper (Cohen, 1995b) a rule induction learning system that produces human readable rules. Using a rule producing machine learning algorithm has the advantage of making it possible for humans to read the rules and try to understand what a mechanical algorithm deems as important features of terms. Produced rules can also be documented and used in other systems.

4.1. Evaluation

Common measures used to evaluate term extraction results are precision, recall and f-score. However, even though the metrics may be the same, applications of the metrics differ between ATR and AKE. As mentioned, in the task of keyword extraction the output is a small list of 5-15 keywords and the performance is measured over this list. Turney (2000) e.g. measures the precision of the first 5, 7, 9, 11, 13, and 15 phrases. In contrast, the length of the output from a ATR system does not have a formal limit, which means that precision and recall numbers cannot be compared between the two tasks.

Zhang et al. (2008) discusses several possible evaluation metrics, noting that many the evaluation metrics used “only measure precision but not recall” and that “they evaluate only a subset of the output”. With the scenario of using ATR output to produce some kind of terminology resource in mind, e.g. as in Merkel et al. (2009), recall is of utmost importance, outranking precision. The reason is that when post-processing the list of extracted terms, it *is possible* to increase the precision of the final result by manually removing incorrect term candidates but, it *is impossible* to increase the recall above the recall of the original list.

Evaluating a subset of the output is not an option in our case as a the design of our experiment does not output a ranked list of term candidates. On the other hand, a subset evaluation method might not be as relevant in ATR as it is in AKE since this evaluation method is precision-oriented, rather than recall-oriented.

4.2. Dataset

The dataset used was provided by Fodina Language Technology and consists of Swedish patent texts grouped by IPC classes. A set of manually validated terms is also provided for each group of patent texts in the data set. The experiments in this paper were run on the smaller A42B subset (*hats; head coverings*) and the larger A61G subset (*transport, personal conveyances, or accommodation specially adapted for patients or disabled persons; operating tables or chairs; chairs for dentistry; funeral devices*). The composition of these data sets is described in tables 1 and 2. For this data set, the term segments refers to a line in the corpus text file which in most cases is a full sentence, but a segment can also be heading or a caption. As can be seen in table 2 there are very few two-word terms, and no three-word terms. This is due to the corpus being in Swedish language where compound nouns are frequently used. For example terms such as ”file manager” or ”file manager window” would both be a single word terms in Swedish: ”filhanterare” and ”filhanterarfönster”).

As the data was made available as a large concatenated text file, document/document collection distributional measures such as *tf-idf* are not possible to calculate. The terms accompanying the patent documents had been previously extracted and were manually validated by domain experts (Merkel et al., 2009).

[1] http://www.compumine.com

Corpus statistics	A42B	A61G
Number of tokens	71761	302027
Number of segments	2929	12684
Number of terms (types)	579	1260

Table 1: Overview of the A42B and A61G document collections.

Term length	A42B	A61G
1 word terms	570	1240
2 word terms	9	20
3 word terms	0	0

Table 2: Composition of the validated term lists. Lack of 2 and 3 word terms is due to use of compounds in the Swedish language.

4.2.1. Feature selection

Based on previous research in ATR, we have chosen to use linguistic features as well as statistical features. Linguistic features were obtained using the commercial tagger Connexor Machinese Syntax[2]. A detailed explanation of these can also be found in (Ahrenberg, 2007). A normalized frequency count was also included. By using a normalized frequency count, the generated rules may also be generalized to other corpora. Furthermore based on previous studies conducted by Foo, a statistical language model of a tokenized general corpus was created and statistics derived from this language model were used as additional features. In the experiments conducted here, the PAROLE corpus[3] was used to build the language model. The motivation behind using a general language language model is to be able to capture how common a word or phrase is in non-domain specific text. All in all, 10 different features were used, as seen in table 3.

Feature	Description
POS	part-of-speech tag
msd	morpho-syntactic description
func	grammatical function
sem	semantic information
nfreq	normalized n-gram frequency in text
zeroprobs	number of tokens with zero probability in given the language model
logprob	the logistic probability value, ignoring unknown words and tokens
ppl1	the geometric average of 1/probability of each token, i.e. perplexity
ppl2	the average perplexity per word

Table 3: List of features used to annotate the examples used in training and test data.

4.3. Preprocessing

The patent texts were provided as one document per subclass, i.e. one document for subclass A42B and one document for subclass A61B. These documents were tagged using Connexor Machinese Syntax and then n-gram extraction was performed which created separate files for each n-gram length while creating the n-gram files. Frequency counts were also done. The files were then annotated with statistics using the language model and finally each n-gram was annotated as a term or a non-term based on the validated term lists.

The result after all preprocessing had been completed was one feature-annotated file for each n-gram length. This file only contains unique examples, with respect to words and linguistic information, i.e. the word "speed" may exist in two unigram rows, but in one row it is tagged as a noun and in the second row it is tagged as a verb.

4.4. Experiments

The key variable in the experiments described in this paper is the positive/negative example ratio used in the training data. We chose five different ratios: 10/90, 30/70, 50/50, 80/20, and 90/10 where the first number is the number of positive examples.

We chose to create separate systems for each n-gram length in our experiments, i.e. one system would create rules for unigrams and another for bigrams. Training and test data used a feature representation which treats tokens in e.g. a bigram as single entities. An alternative is to group tokens into one value. That is, we use a non-aggregated form, e.g. `BIGRAM="SMALL CAR" POS1=A POS2=N` rather than `TEXT="SMALL CAR" POS=A_N`.

In total, 10 experiments per corpus were run, totaling 20 experiments for both corpora. The original plan was to include trigrams, but since no three word terms were included in the term lists, only unigram and bigram experiments were conducted.

After annotating the extracted n-grams according to the process described in section 4.3., 10% randomly chosen example rows from each n-gram set was held back to be used as test data. As positive/negative example ratio is of interest, such properties of the unmodified training set and the test sets are presented in tables 4 and 5. Please note that the ratios reflect unique n-gram data and therefore does not represent an actual term/non-term token ratio of the documents.

The same test set was used for all experiments from within a n-gram group and corpus, e.g. the A42B unigram test set was used for all five unigram experiments conducted on the A42B corpus.

Ripper was run with the following settings for all experiments, `-a given -L 0.4`. The first option `-a given` means that Ripper is forced to use a specified class order. In practice this is a way to force Ripper to produce rules for the "positive" (term) class, even when there are more "negative" (non-term) examples in the training data. The second used option, `-L 0.4` sets the "loss" ratio to 0.4. The loss ratio is the ratio between the cost of a false negative compared to a false positive. A ratio of 0.4 as used in the experiments in this paper, provided a good recall progression in the experiments, i.e. it is possible to produce rules that achieve 100% recall without tipping the scale too much so that all rules sets produce 100% recall. All other settings were left to their default value. Though the set-

[2] http://www.connexor.eu/technology/machinese/machinesesyntax/

[3] http://spraakbanken.gu.se/parole/

tings might have an effect on precision and recall, the point of the experiments are not to find out how the best system performs.

5. Results

The results of the experiments are presented in tables 6 and 7. The general trend is the Ripper system produces rule sets that produce higher recall, the higher the positive/negative ratio is. However, as expected, the precision drops accordingly. The result tables list first unigram experiments then bigram experiments with varying positive/negative example ratios. For example experiment 1-10 refers to a unigram experiment with 10% positive examples in the training data. Experiment 2-30 would be a bigram experiment with 30% positive examples in the training data. The test data set to which the rules are applied are the held back test data described in tables 4 and 5.

As can be seen in the tables, the recall for many of the experiments is 100%. However, this does not mean that all existing terms in the corpus are among the extracted term candidates, it only says that 100% of the validated terms are present. Similarly, precision is relative to the validated term set. However, due to the nature of recall and precision, the real recall can only be equal or lower than the presented recall and the precision can only remain at its current level or improve given a more complete term list. Also, for our experiments, rows with 0 false negatives are the result of rulesets which simply classify all examples as terms. This makes sense from a precision point of view if training data shows that an overwhelming number of examples are terms.

5.1. Rules learned

As it would be unpractical to publish all learned rulesets in this paper, we will only include two sets, one for each n-gram length. These rules are presented in tables 8 and 9. The first rule in table 8 should be read as classify the example as a term (the 'yes' class) `IF` the example frequency (`freq`) is higher or equal to $2.7903e^{-5}$ and its part-of-speech tag is Noun (`n`). The rules are applied in order, i.e. for each example, try to apply a the first rule, if it is not applicable, use the next rule and so on. Interpreting the rules in 8 would produce the following:

- Nouns occurring more than X times are terms
- Nouns above a certain probability given the language model are terms
- All verbs are terms
- All singular nominals are terms
- All plural nominals are terms
- All adjectives are terms
- All singular genitive nouns are terms
- Perfect participles are terms

As can be seen, the rules learned by Ripper use both linguistic and statistical features.

6. Discussion

Comparing the results with e.g. (Zhang et al., 2008) is not possible as our experiment output is not ranked and can therefore not be pruned into a smaller set with higher precision. Including ranking metrics to the n-gram annotation is a good next step for future research. This would also mean that the machine learning algorithm can use this information as well in its learning process.

class	corr	err	condition
yes	832	273	IF freq ≥ 2.7903e-05 pos1 = n .
yes	403	213	IF logprob ≥ 9.66755 pos1 = n .
yes	387	376	IF pos1 = v .
yes	382	482	IF msd1 = sg-nom .
yes	96	75	IF msd1 = pl-nom .
yes	82	97	IF pos1 = a .
yes	25	11	IF pos1 = n msd1 = sg-gen .
yes	16	5	IF pos1 = n logprob ≥ 9.13763 .
yes	37	51	IF pos1 = ad .
no	812	135	IF .

Table 8: Rules learned in experiment A42B 1-50

class	corr	err	condition
yes	11	4	IF func1 = attr pos1 = <cmp> .
yes	22	15	IF pos1 = a logprob ≥ 12.276 .
yes	5	0	IF func1 = attr pos1 = a logprob ≤ 10.8129 logprob ≥ 10.751 .
yes	11	2	IF logprob ≥ 11.8158 pos2 = adv pos1 = v .
yes	4	1	IF func1 = attr pos1 = <ord> .
yes	3	0	IF msd1 = a-nom .
no	509	3	IF .

Table 9: Rules learned in experiment A61G 2-10

Besides using the learned rules to extract term candidates, it is also possible to use feature rich data, i.e. data with many levels of annotation, in combination with machine learning to discover new properties of terms. When it comes to finding rule patterns from huge amounts of data, a rule induction machine learning system such as Ripper can perform many times faster than humans.

Regarding which positive/negative example ratio to use in the training data based on the results presented in this paper, the answer is that it depends on how the output is to be used. In a keyword extraction scenario, high precision is preferred over high recall, which means that a low positive/negative ratio would be recommended. In a scenario where the term candidates will be post-processed by domain experts, a high recall is more important that high precision. In that case a balanced ratio such as 50/50 is recommended as this ratio provides a high precision together with a very high recall (as noted in the Results section, 0 false negatives are the result of a "everything-is-a-term system").

7. Future research

The machine learning approach has been applied to monolingual term extraction in this paper. We have however started working on applying the same approach to bilingual term extraction. Current bilingual term extraction methods, such as presented in (Morin et al., 2007) and (Fan et al., 2009), often rely on monolingual term extraction independently performed on source and target language followed by a term alignment phase between terms in the source and target language (extract-align). There also exist parallel extraction methods that extract terms from aligned texts such as (Merkel and Foo, 2007; Lefever et al., 2009; Foo and Merkel, 2010) (align-extract). A machine learning ap-

	tot	train	train pos	train neg	train pos/neg ratio	test	test pos	test neg	test pos/neg ratio
1-grams	12844	11560	2395	9165	0.261320	1284	485	799	0.607009
2-grams	40221	36199	11	36188	0.000304	4022	9	4013	0.002243

Table 4: A42B experiment data overview

	tot	train	train pos	train neg	train pos/neg ratio	test	test pos	test neg	test pos/neg ratio
1-grams	39243	35319	6239	29080	0.214546	3924	1233	2691	0.458194
2-grams	152853	137568	59	137509	0.000429	15285	30	15255	0.001967

Table 5: A42B experiment data overview

proach presents the opportunity to use a single framework monolingual, bilingual and even multilingual term extraction.

Our current approach uses an n-gram length sensitive feature representation, i.e. a bigram has twice as many linguistic features as a unigram. As a result, for extraction of monolingual terms of length 1 to 3, the machine learning phase has to be split into three separate learning sessions.

Applying this method of representation to bilingual term extraction for terms of length 1 to 3 would require the machine learning phase to be split into nine different sessions (covering 1-1 alignments, 1-2, 1-3 ... 3-3 alignments). Preliminary analysis of bilingually aligned patent texts (English-Swedish) however, indicate that the distribution of aligned phrases according to such a division is very skewed. One way of solving this problem is the change the feature representation to use a single combined feature for all tokens in a phrase rather than separate features for each token in a phrase. This way, the data need not be divided into smaller groups.

8. Conclusion

In this paper, we have shown that machine learning can be used to produce rules which can be used to extract term candidates from a corpus, or more specifically classify n-grams as potential term candidates or not. We have also shown that by using different positive/negative example ratios in the training data, it is possible to govern the kind of rules that are produced. Different kinds of rules may be of interest for different scenarios, so the possibility of dynamically adapting the ruleset to a scenario is promising. Also, we have presented some preliminary results on the use of machine learning for bilingual ATR that indicate that some changes need to be made to to the feature representation scheme to be able to try the approach bilingually.

9. References

Lars Ahrenberg. 2007. Lines 1.0 annotation: Format, contents and guidelines, March.

Sophia Ananiadou. 1994. A methodology for automatic term recognition. In *Proceedings of the 15th conference on Computational linguistics*, pages 1034–1038, Morristown, NJ, USA. Association for Computational Linguistics.

Didier Bourigault. 1992. Surface grammatical analysis for the extraction of terminological noun phrases. In *Proceedings of the 14th conference on Computational linguistics*, pages 977–981, Morristown, NJ, USA. Association for Computational Linguistics.

Jonathan D. Cohen. 1995a. Highlights: language- and domain-independent automatic indexing terms for abstracting. *J. Am. Soc. Inf. Sci.*, 46(3):162–174.

William W. Cohen. 1995b. Fast effective rule induction. In *In Proceedings of the Twelfth International Conference on Machine Learning*, pages 115–123. Morgan Kaufmann.

Béatrice Daille. 1994. Study and implementation of combined techniques for automatic extraction of terminology. In J. Klavans and P. Resnik, editors, *The Balancing Act: Combining Symbolic and Statistical Approaches to Language*, pages 49–66. The MIT Press, Cambridge, Massachusetts.

Xiaorong Fan, Nobuyuki Shimizu, and Hiroshi Nakagawa. 2009. Automatic extraction of bilingual terms from a chinese-japanese parallel corpus. In *IUCS '09: Proceedings of the 3rd International Universal Communication Symposium*, pages 41–45, New York, NY, USA. ACM.

Jody Foo and Magnus Merkel. 2010. Computer aided term bank creation and standardization: Building standardized term banks through automated term extraction and advanced editing tools. In Marcel Thelen and Frieda Steurs, editors, *Terminology in Everyday Life*, pages 163–180. John Benjamins Publishing Company.

Eibe Frank, Gordon W. Paynter, Ian H. Witten, Carl Gutwin, and Craig G. Nevill-Manning. 1999. Domain-specific keyphrase extraction. In *IJCAI'99: Proceedings of the 16th international joint conference on Artificial intelligence*, pages 668–673, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Katerina T. Frantzi, Sophia Ananiadou, and Jun ichi Tsujii. 1998. The c-value/nc-value method of automatic recognition for multi-word terms. In *ECDL*, pages 585–604.

Anette Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. pages 1–8, January.

Anette Hulth. 2004. *Combining Machine Learning and Natural Language Processing for Automatic Keyword Extraction*. Ph.D. thesis, Stockholm University, Department of Computer and Systems Sciences (together with KTH).

Els Lefever, Lieve Macken, and Veronique Hoste. 2009. Language-independent bilingual terminology extraction

experiment	training data	test data	true pos	true neg	false pos	false neg	recall (terms)	precision (terms)
1-10	10183	1284	336	675	124	149	69.28%	73.04%
1-30	7983	1284	424	511	288	61	87.42%	59.55%
1-50	4790	1284	471	465	334	14	97.11%	58.51%
1-80	2993	1284	485	460	339	0	100.0%	58.86%
1 90	2661	1284	485	0	799	0	100.0%	37.77%
2-10	110	4022	5	3855	158	4	55.56%	3.07%
2-30	36	4022	9	3624	389	0	100.0%	2.26%
2-50	22	4022	9	3624	389	0	100.0%	2.26%
2-80	13	4022	9	3126	887	0	100.0%	1.00%
2-90	12	4022	9	0	4013	0	100.0%	0.22%

Table 6: A42B experiment results

experiment	training data	test data	true pos	true neg	false pos	false neg	recall (terms)	precision (terms)
1-10	32311	3924	682	2252	439	551	55.31%	60.84%
1-30	20796	3924	1069	1600	1091	164	86.70%	49.49%
1-50	12478	3924	1233	0	2691	0	100.0%	31.42%
1-80	7798	3924	1233	0	2691	0	100.0%	31.42%
1-90	6932	3924	1228	941	1750	5	99.59%	41.24%
2-10	590	15285	26	14692	563	4	86.67%	4.41%
2-30	196	15285	30	14360	895	0	100.0%	3.24%
2-50	118	15285	30	13586	1669	0	100.0%	1.77%
2-80	73	15285	30	7928	7327	0	100.0%	0.41%
2-90	65	15285	30	0	15255	0	100.0%	0.20%

Table 7: A61G experiment results

from a multilingual parallel corpus. In *EACL '09: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 496–504, Morristown, NJ, USA. Association for Computational Linguistics.

Magnus Merkel and Jody Foo. 2007. Terminology extraction and term ranking for standardizing term banks. In Joakim Nivre, Heiki-Jaan Kaalep, Kadri Muischnek, and Mare Koit, editors, *Proceedings of the 16th Nordic Conference of Computational Linguistics NODALIDA-2007*, Tartu, Estonia. University of Tartu.

Magnus Merkel, Jody Foo, Mikael Andersson, Lars Edholm, Mikaela Gidlund, and Sanna Åsberg. 2009. Automatic extraction and manual validation of hierarchical patent terminology. October.

Emmanuel Morin, Béatrice Daille, Koichi Takeuchi, and Kyo Kageura. 2007. Bilingual terminology mining - using brain, not brawn comparable corpora. In *ACL*.

The parole corpus, at the swedish language bank, university of gothenburg. http://spraakbanken.gu.se/parole/.

J. Ross Quinlan. 1993. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Gerard Salton and Michael J. McGill. 1986. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA.

Peter D. Turney. 2000. Learning algorithms for keyphrase extraction. *Information Retrieval*, 2(4):303–336, May.

Ziqi Zhang, Jose Iria, Christopher Brewster, and Fabio Ciravegna. 2008. A comparative evaluation of term recognition algorithms. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2008/.