# The Use of Topic Representative Words in Text Categorization

*Su Nam Kim*[♠] *and Timothy Baldwin*[♡]

[♠] [♡] Computer Science and Software Engineering
[♡] NICTA VRL
University of Melbourne
Victoria, 3056, Australia
{snkim,tim}@csse.unimelb.edu.au

*Min-Yen Kan*[♣]

[♣] Computer Science
National University of Singapore
Singapore, 117417, Singapore
kanmy@comp.nus.edu.sg

**Abstract** *We present a novel way to identify the representative words that are able to capture the topic of documents for use in text categorization. Our intuition is that not all word $n$-grams equally represent the topic of a document, and thus using all of them can potentially dilute the feature space. Hence, our aim is to investigate methods for identifying good indexing words, and empirically evaluate their impact on text categorization. To this end, we experiment with five different word sub-spaces: title words, first sentence words, keyphrases, domain-specific words, and named entities. We also test TF·IDF-based unsupervised methods for extracting keyphrases and domain-specific words , and empirically verify their feasibility for text categorization. We demonstrate that using representative words outperforms a simple 1-gram model.*

**Natural Language Techniques and Documents, Text Categorization**

## 1 Background and Motivation

*Automatic text categorization* is the task of classifying documents into a set of predefined categories. It is one of the more heavily researched areas in natural language processing (NLP) due to its immediate applicability in applications such as text filtering [1], word sense disambiguation [11] and automated authorship attribution and genre classification [8].

The conventional approach to text categorization utilizes supervised machine learners such as support vector machines (SVMs) and Maximum Entropy (ME) models, and represents each document as a bag of word $n$-grams [40, 14, 10]. Empirically, SVMs have been shown to be superior to other machine learning techniques such as Naive Bayes (NB), Rocchio and decision trees over a range of tasks [40, 10].

While the predominance of research in text categorization is on machine learning models, there has also been significant research on feature extraction [4, 2, 24, 21] and feature weighting/selection [18, 41, 7]. While the majority of research has used simple $n$-grams to represent documents [4], this has been expanded in various ways, including word clusters [2], complex nominals [24], words from automatically extracted sentences [21], and title words/keyphrases(or keywords) [13]. Similarly, while most research has used simple term weighting (*TF* and/or *TF·IDF* variants), some have used attributes such as mutual information [18], chi-square [41], and gain ratio [7] to weight and/or select features.

Our interest is in the impact of different term types on text categorization. Our intuition is that not all word $n$-grams equally represent the topic of a document, and thus using all of them can potentially dilute the feature space. Hence, our aim is to investigate methods for identifying good indexing words, and empirically evaluate their impact on text categorization. To find representative topic words, we tested five different word groups: *title words, first sentence words, domain-specific words, named entities*, and *keyphrases*. *Title words* and *first sentence words* are based on the notion of *document zoning*. *Domain-specific words* and *named entities*, on the other hand, are typified as occurring with markedly-high occurrence in documents of particular domains. Finally, keyphrases are representative words, as identified by dedicated methods such as [12] and [36]. We also test combining the different term types with conventional terms $n$-grams.

A secondary area of interest in this research is exploration of the utility of unsupervised term extraction methods. As a result, we are particularly interested in the utility of unsupervised keyphrase and domain-specific word extraction methods on text categorization.

## 2 Zone-based Term Extraction

Our first term extraction method is based on document zoning, i.e. the extraction of terms based on the document structure. A common approach in keyphrase extraction and topic detection is to use titles as a representation of the document topic. For example, [26] showed that sentences in particular article sections, such as the introduction and conclusion, contain more keyphrases in scientific articles.

**Proceedings of the 14th Australasian Document Computing Symposium, Sydney, Australia, 4 December 2009.**

In our work, we drew on methods such as [21] in extracting important sentences from documents based on the simple heuristic that the title and first sentence often contains key facts about the news story. From these observations, we select the *title words* and *first sentence words* as candidate terms. In each case, we extract out the component 1-grams, to minimize reliance on parsing or manual processing. We also filter terms by their combined occurrence in the document set, selecting only those terms which occur with frequency $\geq 1$, 2 or 3. The final number of title words is 8,622, 3,878, and 2,357, for cutoffs of 1, 2 and 3, respectively, and the corresponding number of first sentence words is 11,565, 5,819, and 3,905, respectively. These numbers are based on the evaluation data described in Section 6.

## 3 Keyphrases

*Keyphrases* are simplex (i.e. 1-gram) nouns or noun phrases that represent the key ideas of the document. Keyphrases can serve as a condensed summary of the document and also as high-quality index terms. In the past, the majority of keyphrase studies have used three types of statistics to extract keyphrases: (1) document co-occurrence, i.e. *TF·IDF*-style statistics relating keyphrases to their relative co-occurrence across documents [12, 26]; (2) keyphrase co-occurrence, i.e. the extent to which keyphrases occur together in the same documents [37]; and (3) term co-occurrence, i.e. local contiguity of terms in keyphrases [28].

We quickly summarize related work first. `KEA` [12] is a very simple and popular keyphrase extraction and indexing tool. It uses two main features: *TF·IDF* to capture document co-occurrence, and *distance* to signify the relative locality of keyphrase occurrences within documents. These features have been broadly used in keyphrase extraction, e.g. by [37] in addition to keyphrase co-occurrence. [26] extended the basic `KEA` approach by applying linguistic features such as document zones. `GenEx` [36] uses more syntactic features, such as document positions and stemming. [3] uses head noun-based heuristics. [35] use modelling based on information loss between preceding and proceeding document extents. `Textract` [28] ranks keyphrase candidates by their degree of domain-specificity and term cohesion in a text analysis system. [38] uses information from clustered documents for keyphrase extraction over single documents.

### 3.1 Unsupervised Keyphrase Extraction

As keyphrases are known to be representative of document topics, it is also natural to use them as terms for document categorization. Hulth and Megayesi [13] used a supervised keyphrase extraction method, seeded with 500 abstracts annotated with keyphrases. To avoid documents without keyphrases, they controlled the number of keyphrases to between 3 and 12.

While supervised techniques work well, they require manually-built annotated corpora, which has implications both in terms of resource creation and domain adaptability. We are interested in minimizing such efforts, and thus committed to using unsupervised or minimally-supervised methods. To the best of our knowledge, very few unsupervised keyphrase extraction methods exist. Therefore, we used the features used in `KEA` to build our own unsupervised keyphrase extractor. That is, we use *TF·IDF* and *first position*, i.e. the inverse of the offset from the start of the document, such that documents which occur earlier in the document are preferred as keyphrase candidates. First, we calculate the score for each candidate as shown in (1), combining *TF·IDF* and first position.

| Word set | T1(.02) | T2(.04) | T3(.06) |
|---|---|---|---|
| original | 7,889 | 5,733 | 4,497 |
| 1+NP | 25,343 | 15,257 | 10,679 |

Table 1: Number of collected keyphrases

$$Score = \textit{TF·IDF} + (1 - \frac{\textit{first position of } W_i}{\textit{\# of total terms}}) \quad (1)$$

We then extract the top-$N$ candidates as keyphrases. In other keyphrase extraction research, $N$ has typically been set to 15, but in our case, we decided to experiment with different thresholds. This is because the documents used in text categorization testbeds are short, and thus result in comparatively few keyphrase candidates. We selected thresholds by examining the score drop. Specifically, we set the threshold to the point at which the number of domain-specific terms gained at the current similarity value is no more than a fixed proportion (e.g. 2%) of keyphrases previously selected. Due to this use of threshold, our keyphrase extractor did not assign any keyphrases for a few documents.

Keyphrases can be either simplex nouns or NPs. [13] found that breakdown-keyphrases (i.e. all unigrams contained within a keyphrase) performed better for text categorization. Hence, we also convert keyphrases into their component unigrams. However, we observed that whole keyphrases are often better descriptors of the document topic (e.g. *import goods* vs. *goods*). Thus, we tested another set, called 1+NP, which combines 1-grams with the original keyphrases.

Table 1 shows the number of collected keyphrases for the entire document collection (see Section 6) at different threshold settings, for both the original keyphrases and 1+NP. Figure 1 additionally shows the proportion of documents containing different numbers of keyphrases for the three thresholds.

To assess the quality of our unsupervised keyphrase extractor, we sampled 100 documents from the training data and had two human annotators manually assign keyphrases to 50 documents each. The total number of manually-assigned keyphrases in the 100 sample documents was $1,486$. Performance is shown in Table 2.

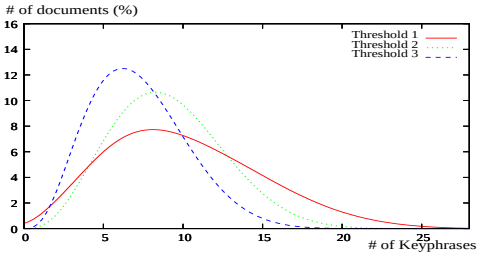


Figure 1: Proportion of documents assigned differing numbers of keyphrases

| | Precision | Recall | Fscore |
|---|---|---|---|
| T1(.02) | 9.76% | 23.85% | 13.85% |
| T2(.04) | 15.32% | 15.62% | 15.47% |
| T3(.06) | 21.02% | 10.86% | 14.32% |

Table 2: Performance of keyphrase extraction

## 4 Domain-Specific Terms

*Automatic domain-specific term extraction* is a classification process where the terms are categorized using a set of predefined domains with supervised machine learning models. It has been studied for application in areas such as keyphrase extraction [12, 38] and word sense disambiguation [19].

Much of the work has been carried out using supervised machine learning techniques in the context of term categorization and/or text mining. [9] focused on simplex terms using corpus comparison, and verified the collected data using automatic and manual validation. [31] projected the categorized terms onto a predefined set of semantic domains exploiting web knowledge, and used the context to map the terms onto domains. [29] proposed an unsupervised method for extracting domain-specific terms, and used them to check word and keyword error rates.

In this paper, we test two unsupervised domain-specific word extraction approaches, drawing on work in the context of keyphrase extraction [16]. The first one (**D1**) is based on simple *TF·IDF*. The second method (**D2**) was proposed by [29], and is based on the difference in *TF* for a given domain relative to other domains, based on:

$$\textbf{D2} = domain\_specificity(w) = \frac{\frac{c_d(w)}{N_d}}{\frac{c_g(w)}{N_g}} \qquad (2)$$

where $c_d(w)$ and $c_g(w)$ denote the number of occurrences of term $w$ in the domain text and general document collection, respectively. $N_d$ and $N_g$ are the numbers of terms in the domain corpus and in the general corpus, respectively. If term $w$ does not occur in the general corpus, then $c_g(w)$ is set to 1; otherwise it is set to the highest count in the general corpus.

We use the same thresholding method for the two methods as described in Section 3.1.

| Method | Term set | T1(.02) | T2(.04) | T3(.06) |
|---|---|---|---|---|
| D1 | original | 2,918 | 1,573 | 1,157 |
| | 1+NP | 3,969 | 1,918 | 1,344 |
| D2 | original | 3,692 | 2,759 | 2,368 |
| | 1+NP | 7,169 | 5,021 | 4,215 |

Table 3: Number of collected domain-terms words

| | Overlap | D1 | D2 |
|---|---|---|---|
| T1 | 1,612 | 55.24% | 43.67% |
| T2 | 593 | 37.70% | 21.49% |
| T3 | 404 | 34.92% | 17.06% |

Table 4: Overlap between domain-specific words collected by D1 and D2

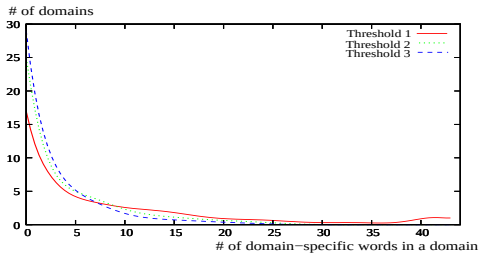


Figure 2: Number of domains containing differing numbers of domain-specific terms for D1

Table 3 details the number of terms and 1+NP extracted by D1 and D2 over the document collection described in Section 6, over three different threshold values. We also calculated the overlap in terms extracted by the two methods, and report the numbers in Table 4. The numbers in the second and third columns show the portion of terms extracted by the D1 and D2, respectively, which overlap with terms extracted by the second method.

The number of domains containing differing numbers of terms is shown in Figures 2 and 3. D1 produced less domain-specific words in total (as shown in Table 3), but the keyphrases are better distributed across the domains.

In separate research, we manually evaluated the terms extracted by the two methods, and found that D1 marginally outperformed D2 [16].

## 5 Named Entities

*Named entity recognition* is the task of identifying atomic elements in a document which belong to predefined categories such as location, person, and organization. It has been applied to contexts including Question-Answering (QA) [23] and information retrieval [34]. The standard approach is based on structured classification methods such as hidden Markov models (HMMs) or conditional random fields (CRFs). Recently, research has focused on semi-

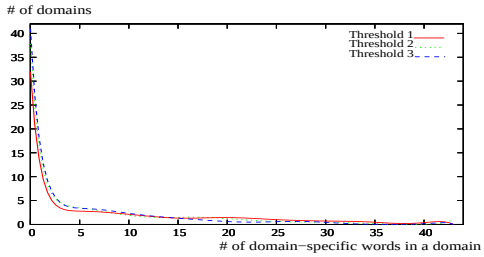


Figure 3: Number of domains containing differing numbers of domain-specific terms for D2

| Length | F1($f \geq 1$) | F2($f \geq 2$) | F3($f \geq 3$) |
|---|---|---|---|
| original | 11,431 | 6,538 | 4,650 |
| 1+NP | 23,440 | 9,883 | 6,234 |

Table 5: Number of extracted named entities

supervised [27] and/or unsupervised approaches [5] to named entity recognition.

The relevance of named entities (NEs) to this research is that we expect they will be indicative of document domains. For example, *Gulf* and *Kuwait* often occur in the domain of *oil* and not other domains. Thus, we trial named entities as a term type in text categorization.

We experiment exclusively with the named entity recognition software of the University of Illinois at Urbana-Champaign (`UIUC NER`).[1] `UIUC NER` makes extensive use of non-local features and external knowledge resources (i.e. gazetteers extracted from Wikipedia), as well as semi-supervised learning. It identifies four entity types (i.e. person, location, organization and miscellaneous), and is reported to have achieved 90.80 F1-score over the CoNLL-03 NER shared task

Table 5 shows the number of named entities extracted by `UIUC NER` over our document collection (see Section 6). We used three different frequency cutoffs to select the candidate NEs ($f_{NE} \geq 1, 2, 3$), and once again experimented with both the original NEs and the 1+NP method of breaking down the NEs.

## 6 Text Categorization

We now describe our integrated approach for performing text categorization, incorporating the various extracted term types from the preceding sections.

As our dataset, we use the Reuters newswire corpus, with 21,450 articles from 1987, spanning 135 topics. The number of articles with no category label, one label and multiple labels are 31%, 57% and 12%, respectively. This dataset has been used widely for text categorization research. In particular, we use the *Modified Lewis Split*, comprising 7,771 training and 3,019 test documents across 90 domains.[2]

In preprocessing, we performed part-of-speech (POS) tagging using the *Lingua* POS tagger, and POS-sensitive lemmatization using `morpha` [22].[3] Then we built classifiers using SVM$^{light}$,[4] with *TF·IDF* term weighting in an attempt to generate as competitive as possible a text categorization system.

As our benchmark, we use 1-grams with a frequency cutoff of 1, 2 and 3 (i.e. all terms occurring less than $N$ times are ignored), along with stopping. The best results were achieved for a frequency cutoff of 3, with a micro-averaged F-score of 78.54%.

Table 6 shows the text categorization performance of the various term extraction methods, organized into four groups: (1) individual extraction methods; (2) the combination of all extraction methods; (3) the combination of individual extraction methods with 1-grams; and (4) the combination of all extraction methods with 1-grams. In each case, we report the micro-averaged precision, recall and F-score ($\beta = 1$) for the given method over the test data. All values which surpass the benchmark performance (F3) at a level of statistical significance (based on approximate randomisation, $p < 0.05$) are indicated in **bold**. In Table 6, F1, F2 and F3 refer to the three frequency cutoffs used for title words, first sentence words and named entities ($f \geq 1, 2, 3$), while T1, T2 and T3 refer to the three thresholds used for keyphrases and domain-specific words. We also present the performance over the top-10 topics in Table 7.

## 7 Text Categorization Results

Looking first at the individual methods (the top section of Table 6), we notice that only keyphrases were able to surpass the performance of the benchmark, closely followed by title and first sentence words, then named entities, and finally domain-specific terms. Almost no difference was observed between using the original terms extracted by each of the methods, and combining the original terms with their unigram components (1+NP). In general, the standalone methods tended to do better in terms of both precision and recall for lower cutoff/threshold values, that is larger numbers of noisier terms tended to boost performance across the board.

When we combine all five term extraction methods (considering D1 and D2 separately), the results exceed those of the benchmark in all cases for the lowest threshold/cutoff values, and in select cases for higher values. None of these gains were found to be statistically significant, and yet the result is encouraging as the best of the combined methods outperforms the best of the standalone methods,

[1] `http://l2r.cs.uiuc.edu/~cogcomp/asoftware.php?skey=FLBJNE`

[2] `http://www.daviddlewis.com/resources/testcollections/reuters21578/`

[3] The only use we made of the POS tags was in lemmatization.

[4] `http://svmlight.joachims.org/svm_multiclass.html`

| Word | Length | F1/T1 Prec. | F1/T1 Recall | F1/T1 Fscore | F2/T2 Prec. | F2/T2 Recall | F2/T2 Fscore | F3/T3 Prec. | F3/T3 Recall | F3/T3 Fscore |
|---|---|---|---|---|---|---|---|---|---|---|
| Benchmark | 1 | 87.15% | 70.26% | 77.80% | 87.48% | 70.53% | 78.09% | 87.98% | 70.93% | 78.54% |
| Title (T) | 1 | 87.48% | 70.53% | 78.09% | 87.58% | 70.61% | 78.18% | 87.58% | 70.61% | 78.18% |
| First (F) | 1 | 87.58% | 70.61% | 78.18% | 87.48% | 70.53% | 78.09% | 87.35% | 70.42% | 77.98% |
| Keyphrase (K) | 1 | 88.01% | 70.95% | 78.57% | 87.45% | 70.50% | 78.07% | 87.68% | 70.69% | 78.27% |
| | 1+NP | 87.78% | 70.77% | 78.36% | 87.65% | 70.66% | 78.24% | 87.65% | 70.66% | 78.24% |
| Domain (D1) | 1 | 86.26% | 69.54% | 77.00% | 85.70% | 69.08% | 76.50% | 83.44% | 67.27% | 74.49% |
| | 1+NP | 86.26% | 69.54% | 77.00% | 85.70% | 69.08% | 76.50% | 83.44% | 67.27% | 74.49% |
| Domain (D2) | 1 | 84.67% | 68.26% | 75.58% | 82.78% | 66.73% | 73.90% | 81.75% | 65.91% | 72.98% |
| | 1+NP | 84.67% | 68.26% | 75.58% | 82.78% | 66.73% | 73.90% | 81.75% | 65.91% | 72.98% |
| NE (N) | 1 | 86.16% | 69.46% | 76.91% | 85.53% | 68.95% | 76.35% | 84.87% | 68.42% | 75.76% |
| | 1+NP | 86.32% | 69.59% | 77.06% | 85.53% | 68.95% | 76.35% | 85.17% | 68.66% | 76.03% |
| T+F+K+D1+N | 1 | 87.98% | 70.93% | 78.54% | 87.91% | 70.87% | 78.48% | 87.78% | 70.77% | 78.36% |
| | 1+NP | 88.11% | 71.03% | 78.66% | 87.72% | 70.71% | 78.30% | 87.91% | 70.87% | 78.48% |
| T+F+K+D2+N | 1 | 88.05% | 70.98% | 78.60% | 87.95% | 70.90% | 78.51% | 88.01% | 70.95% | 78.57% |
| | 1+NP | 88.15% | 71.06% | 78.69% | 88.08% | 71.01% | 78.63% | 88.25% | 71.14% | 78.77% |
| B3+Title | 1 | 87.72% | 70.71% | 78.30% | 87.85% | 70.82% | 78.42% | 87.55% | 70.58% | 78.15% |
| B3+First | 1 | 87.78% | 70.77% | 78.36% | 87.62% | 70.63% | 78.21% | 87.82% | 70.79% | 78.39% |
| B3+Keyphrase | 1 | 88.18% | 71.09% | 78.72% | 87.85% | 70.82% | 78.42% | 88.05% | 70.98% | 78.60% |
| | 1+NP | 88.31% | 71.19% | 78.83% | 88.38% | 71.25% | **78.89%** | 88.15% | 71.06% | 78.69% |
| B3+D1 | 1 | 87.95% | 70.90% | 78.51% | 88.08% | 71.01% | 78.63% | 87.95% | 70.90% | 78.51% |
| | 1+NP | 87.95% | 70.90% | 78.51% | 88.08% | 71.01% | 78.63% | 87.95% | 70.90% | 78.51% |
| B3+D2 | 1 | 87.45% | 70.50% | 78.07% | 87.32% | 70.59% | 77.95% | 87.68% | 70.69% | 78.27% |
| | 1+NP | 87.45% | 70.50% | 78.07% | 87.32% | 70.59% | 77.95% | 87.68% | 70.69% | 78.27% |
| B3+NE | 1 | 87.58% | 70.61% | 78.18% | 87.68% | 70.69% | 78.27% | 87.98% | 70.93% | 78.54% |
| | 1+NP | 87.58% | 70.61% | 78.18% | 87.65% | 70.66% | 78.24% | 87.45% | 70.50% | 78.07% |
| B3+T+F+K+D1+N | 1 | 88.28% | 71.17% | 78.80% | 88.31% | 71.19% | 78.83% | 88.25% | 71.14% | 78.77% |
| | 1+NP | 88.44% | 71.30% | **78.95%** | 88.15% | 71.06% | 78.69% | 88.21% | 71.11% | 78.75% |
| B3+T+F+K+D2+N | 1 | 88.31% | 71.19% | 78.83% | 88.28% | 71.17% | 78.80% | 88.48% | 71.33% | **78.98%** |
| | 1+NP | 88.44% | 71.30% | **78.95%** | 88.38% | 71.25% | **78.89%** | 88.48% | 71.33% | **78.98%** |

Table 6: Performance of text categorization

| Benchmark (F3) | Individual | Individual+1-grams | All candidates | All candidates+1-grams |
|---|---|---|---|---|
| 89.55% | 89.59% | 89.96% | 90.02% | 90.07% |

Table 7: Performance over the top-10 topics

suggesting that there is complementarity between the term extraction methods. Comparing D1 and D2, our simple *TF·IDF*-based unsupervised term extraction method is marginally superior to D2 (the method of [16]).

Next, when we combine the individual methods with the terms from the benchmark method, the results improve uniformly, with the best-performing method (keyphrases with 1+NP terms) surpassing the benchmark method at a level of statistical significance. This indicates that keyphrases, as extracted using our adaptation of KEA, can successfully complement simple 1-grams in text categorization.

Finally, when we combine the benchmark term representation with all of the term extraction methods, we again achieve statistically significant gains almost 50% of the time, once again pointing to the utility of term extraction methods in text categorization applications. Comparing these results with those for the standalone term extraction methods combined with the benchmark system, the full set of five methods is not able to improve significantly beyond the performance of keyphrase extraction with the benchmark system.

Looking to the results over the top-10 topics, we find a similar trend, with keyphrases producing the best standalone performance, and all term extraction methods combined with 1-grams producing the best overall performance.

## 8 Conclusions

In this work, we evaluated the impact on text categorization of five representative term extraction methods, namely title words, first sentence words, keyphrases, domain-specific words, and named entities. We used the output of the different methods, either individually or in combination, as the source of terms for text categorization, and verified that we were able to achieve statistically significant improvements over a benchmark text categorization method using either keyphrase extraction in combination with the benchmark term representation, or the combination of all term extraction methods, again in combination with the benchmark term representation. On the basis of this, we concluded that keyphrases were the pick of the terms experimented with, but also that there is complementarity between the different term types.

## Acknowledgements

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

## References

[1] G. Amati and D. DAloisi and V. Giannini and F. Ubaldini, A framework for filtering news and managing distributed data, Journal of Universal Computer Science, 1997, 3(8), pp. 1007–1021.

[2] L.D. Barker and A.K. McCalluma, Distributional clustering of words for text categorization, In Proceedings of 21st ACM International Conference on Research and Development in Informatoin Retrieval, 1998, pp.96–103.

[3] K. Barker and N. Cornacchia, Using noun phrase heads to extract document keyphrases, In Proceedings of the 13th Biennial Conference of the Canadian Society on Computational Studies of Intelligence: Advances in Artificial Intelligence, 2000, pp. 40–52.

[4] W.B. Cavnar and J.M. Trenkle, N-gram-based text categorization, In Proceedings of SDAIR, 1994, pp. 161–175.

[5] M. Collins and Y. Singer, Unsupervised Models for Named Entity Classification, In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, 1999, pp. 100–110.

[6] D. Okanohara and Y. Miyao and Y. Tsuruoka and J. Tsujii, Improving the Scalability of Semi-Markov Conditional Random Fields for Named Entity Recognition, In Proceedings of COLING/ACL, 2006, pp. 465–472.

[7] F. Debole and F. Sebastiani, Supervised term weighting for automated text categorization, In 18th ACM Symposium on Applied Computing, 2003, pp.784–788.

[8] J. Diederich and J. Kindermann and E. Leopold and G. Paass, Authorship attribution with support vector machines, Applied Intelligence, 2003, 19(1/2), pp.109–123.

[9] P. Drouin, Detection of Domain Specific Terminology Using Corpora Comparison, In Proceedings of the 4th LREC, 2004, pp. 79–82.

[10] S. Dumais and J. Platt and D. Heckerman and M. Sahami, Inductive learning algorithms and representations for text categorization, In Proceedings of CIKM, 1998, pp. 148–155.

[11] G. Escudero and L. Marquez and G. Rigau, Boosting applied to word sense disambiguation, In Proceedings of 11th European Conference on Machine Learning, 2000, pp. 129–141.

[12] E. Frank and G.W. Paynter and I. Witten and C. Gutwin and C.G. Nevill-Manning, Domain Specific Keyphrase Extraction, In Proceedings of the 16th IJCAI, 1999, pp. 668–673.

[13] A. Hulth and B. Megayesi, A Study on Automatically Extracted Keywords in Text Categorization, In Proceedings of the 21st COLING/ACL, 2006, pp. 537–544.

[14] T. Joachims, Text categorization with support vector machines: Learning with many relevant features, In Proceedings of ECML, 1998, pp. 137–142.

[15] M. Kida, M. Tonoike, T. Utsuro and S. Sato, Domain Classification of Technical Terms Using the Web, Systems and Computers, 2007, 38(14), pp. 2470–2482.

[16] S. Kim, T. Baldwin and M-Y. Kan, An Unsupervised Approach to Domain-Specific Term Extraction, In Proceedings of the Australasian Language Technology Workshop 2009, to appear.

[17] Y. Ko and J. Park and J. Seo, Improving text categorization using the importance of sentences, Information Processing and Management, 2004, 40(1), pp. 65–79.

[18] D.D. Lewis, An evaluation of phrasal and clustered representations on a text categorization task, In 15th ACM International Conference on Research and Development in Informaton Retrieval, 1992, pp. 37–50.

[19] B. Magnini and C. Strapparava and G. Pezzulo and A. Gliozzo, The role of domain information in word sense disambiguation, Natural Language Engineering, 2002, 8(4), pp. 359–373.

[20] Y. Matsuo and M. Ishizuka, Keyword Extraction from a Single Document using Word Co-occurrence Statistical Information, International Journal on Artificial Intelligence Tools, 2004, 13(1), pp. 157–169.

[21] R. Mihalcea and S. Hassan, Using the essence of texts to improve document classification, In Proceedings of RANLP, 2005.

[22] G. Minnen and J. Carroll and D. Pearce, Applied morphological processing of English, Natural Language Engineering, 2001, 7(3), pp. 207–223.

[23] D. Molla and M. van Zaanen and D. Smith, Named Entity Recognition for Question Answering, In Proceedings of ALTW, 2006, pp. 51–58.

[24] A. Moschitti and R. Basili, Complex linguistic features for text classification, In Proceedings of 26th European Conference on Information Retrieval Research, 2004, pp.181–196.

[25] D. Nadeau and P.D. Turney and S. Matwin, Unsupervised Named-Entity Recognition: Generating Gazetteers and Resolving Ambiguity, In cogprints, 2006, pp. 266–277.

[26] T. Nguyen and M.Y. Kan, Key phrase Extraction in Scientific Publications, In Proceeding of International Conference on Asian Digital Libraries, 2007, pp. 317-326.

[27] S. Pakhomov, Semi-Supervised Maximum Entropy Based Approach to Acronym and Abbreviation Normalization in Medical Texts, In Proceedings of 40th ACL, 2002, pp. 160–167.

[28] Y. Park and R.J. Byrd and B. Boguraev, Automatic Glossary Extraction Beyond Terminology Identification, In Proceedings of COLING, 2004, pp. 48–55.

[29] Y. Park and S. Patwardhan and K. Visweswariah and S.C. Gates, An Empirical Analysis of Word Error Rate and Keyword Error Rate, In Proceedings of International Conference on Spoken Language Processing, 2008, pp. 2070–2073.

[30] L. Ratinov and D. Roth, External Knowledge and Non-local Features in Named Entity Recognition, In Proceedings of NAACL, 2009.

[31] L. Rigutini and E. Di Iorio and M. Ernandes and M. Maggini, Automatic term categorization by extracting knowledge from the Web, In Proceedings of 17th ECAI, 2006, pp. 531–535.

[32] G. Salton and A. Wong and C.S. Yang, A vector space model for automatic indexing, Communications of the ACM, 1975, 18(11), pp. 61–620.

[33] F. Sebastiani, Machine learning in automated text categorization, ACM Computering Surveys, 2002, 34(1), pp. 1–47.

[34] S. Sekine and K. Sudo and C. Nobata, Extended Named Entity Hierarchy, In Proceedings of LREC, 2002.

[35] T. Tomokiyo and M. Hurst, A Langauge Model Approach to Keyphrase Extraction, In Proceedings of ACL Workshop on Multiword Expressions, 2003, pp.33–40.

[36] P. Turney, Learning to Extract Keyphrases from Text, In National Research Council, Institute for Information Technology, Technical Report ERB-1057, 1999.

[37] P. Turney, Coherent keyphrase extraction via Web mining, In Proceedings of the 18th IJCAI, 2003, pp. 434–439.

[38] X. Wan and J. Xiao, CollabRank: towards a collaborative approach to single-document keyphrase extraction, In Proceedings of COLING, 2008, pp. 969–976.

[39] I. Witten and G. Paynter and E. Frank and C. Gutwin and G. Nevill-Manning, KEA:Practical Automatic Key phrase Extraction, In Proceedings of the fourth ACM conference on Digital libraries, 1999, pp.254–256.

[40] Y. Yang and X. Liu, A re-examination of text categorization methods, In Proceedings of SIGIR, 1997, pp. 42–49.

[41] Y. Yang and J.O. Pedersen, A comparative study on feature selection in text categorization, In Proceedings of 14th International Conference on Machine Learning, 1997, pp. 412–420.