Ontology-Based Feature Transformations: A Data-Driven Approach

Filip Ginter, Sampo Pyysalo, Jorma Boberg, Jouni Järvinen, and Tapio Salakoski

Turku Centre for Computer Science and Department of Information Technology, University of Turku, Lemminkäisenkatu 14, 20520 Turku, Finland
firstname.lastname@it.utu.fi

Abstract. We present a novel approach to incorporating semantic information to the problems of natural language processing, in particular to the document classification task. The approach builds on the intuition that semantic relatedness of words can be viewed as a non-static property of the words that depends on the particular task at hand. The semantic relatedness information is incorporated using feature transformations, where the transformations are based on a feature ontology and on the particular classification task and data. We demonstrate the approach on the problem of classifying MEDLINE-indexed documents using the MeSH ontology. The results suggest that the method is capable of improving the classification performance on most of the datasets.

1 Introduction

Many natural language processing tasks can benefit from information about semantic relatedness of words. For example, the methods for information retrieval and text classification tasks can be extended to capture information about words that are lexically distinct but semantically related. This is in contrast with the common bag-of-words representation of text where no semantic relatedness information is captured. Information on semantic relatedness of words can be beneficial in at least two practical ways. Combining the related cases that would be distinct in the standard bag-of-words representation may result in a better predictor, for example, by yielding more accurate maximum-likelihood estimates in probabilistic methods such as the naive Bayes classifier. Further, words that are very rare or even unseen during training, but are closely semantically related to some more frequent word, can be used as a source of information.

Semantic networks such as WordNet[1] and UMLS[2] are obvious sources of semantic knowledge about words. The semantic networks are usually represented as graphs with nodes representing words and edges representing semantic relationships such as synonymy, hypernymy, and meronymy, for example.

[1] http://www.cogsci.princeton.edu/~wn/
[2] http://www.nlm.nih.gov/research/umls/

One way to incorporate the information on semantic relatedness of words is to define a quantitative measure that can be used in various classification and clustering techniques. The need for such a quantitative measure has given rise to various techniques that measure pairwise word semantic relatedness based on semantic networks.

The approach of Rada and Bicknell (1989) defines the strength of the relationship between two words in terms of the minimum number of edges connecting the words in the semantic network graph. Resnik (1995) argues that the semantic distance covered by single edges varies and employs a corpus-based method for estimating the distance of related concepts. Budanitsky (1999) presents an application-oriented evaluation of these two and three other methods. It should be noted that these methods aim to measure the strength of the pairwise word relationship as a static property of the words, that is, the strength of the relationship is defined independently of the task at hand.

In this paper, we devise and investigate techniques that are based on the intuition that an optimal measure of semantic relatedness is not a static property of words, but depends also on the problem at hand. To illustrate the intuition, let us consider the task of text classification and the two words "mouse" and "human". For many text classification tasks, it would be beneficial to consider "mouse" and "human" to be relatively distant, but in case of the hypothetical classification task where the goal is to distinguish between documents about eucaryotes and procaryotes, it might be beneficial to consider "mouse" and "human" similar or even synonymous. Conversely, the two words "wheat" and "oat" would typically be considered closely related, but, for example, in the Reuters-21578 classification dataset,[3] where the two words define distinct classes, it would be beneficial to consider the words unrelated. Relating features in a task-specific manner has also been considered by Baker and McCallum (1998), who introduce a feature clustering method with a primary focus on dimensionality reduction. However, their method is not governed by semantic networks, but it is based on the distribution of class labels associated with each feature.

Instead of defining a quantitative measure of the strength of semantic relationship between words, we incorporate the semantic information in the form of transformations based on the hierarchical ontology that underlies the words. The relations encoded in the hierarchy are the starting point of the proposed method. The method then operates on a given training set for a given problem, and it attempts to identify elementary transformations of the features that are beneficial to the performance of a machine learning method for the problem. Roughly, each transformation decides on the relatedness or unrelatedness of a set of words. In the mouse vs. human example above, the method would be expected to relate the words "mouse" and "human" only if such a step improves the performance of the machine learning method on the task.

We apply the method to a classification of MEDLINE-indexed[4] documents, where each document is annotated with a set of terms from the MeSH ontol-

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

[4] http://www.nlm.nih.gov/

ogy[5]. However, the method is applicable to any problem where the features are organized in a hierarchy and a measure of performance can be defined.

The paper is organized as follows. In Section 2 we define the necessary concepts and describe the method. Section 3 describes an application of the method to biomedical literature mining based on the MeSH ontology. The empirical results and possible future directions are discussed in Section 4, and Section 5 concludes the paper.

2 Feature transformations

In this section we define a feature hierarchy in the form of a tree and present some of the possible elementary feature mappings based on the hierarchy.

2.1 Feature hierarchy

Let F be a finite set of possible features that are organized into a "is a" concept hierarchy in the form of a tree. Let $a, b \in F$ be features. If a is a child of b, denoted as $a \prec b$, we say that a is a *direct specialization* of b and b is a *direct generalization* of a. If a is a descendant of b, denoted as $a \prec^* b$, we say that a is a *specialization* of b and b is a *generalization* of a.

Let further $\mathcal{G}(b) = \{a \mid b \prec^* a\}$ be the set of all generalizations of b. Similarly, let $\mathcal{S}(a) = \{b \mid b \prec^* a\}$ be the set of all specializations of a. We say that a is *most general* if it is the root of the hierarchy, that is, $\mathcal{G}(a) = \emptyset$. Similarly, we say that b is *most specific* if it is a leaf of the hierarchy, that is, $\mathcal{S}(b) = \emptyset$.

2.2 Elementary transformations of feature multisets

Each document is represented as a multiset $X \subseteq F$ of features extracted from the document. Before X is passed to a text classifier, it undergoes a transformation, which may remove some features from the multiset, or replace some features with a multiset of (possibly different) features. The feature multiset transformations are independent of the classification method used for the data, since the classifier is applied only after the features were transformed.

In order to search through the space of possible feature multiset transformations, we define a set of elementary transformations, where each elementary transformation is a feature multiset mapping $2^F \to 2^F$. A locally optimal transformation is obtained as a composition of several elementary transformations. In the following, we consider some of the possible elementary transformations.

Generalization to a feature The generalization to a feature transformation is parametrized by a feature $a \in F$ and it causes all features belonging to $\mathcal{S}(a)$ (that is, features more specific than a) to be replaced by the feature a, in other words, the whole subtree under the feature a is "folded" up to the feature a.

[5] http://www.nlm.nih.gov/mesh/meshhome.html

The transformation causes all features belonging to $\mathcal{S}(a)$ to be treated as full synonyms of a:

$$G_a(X) = \bigcup_{x \in X} g_a(x)\,, \text{ where} \tag{1}$$

$$g_a(x) = \begin{cases} \{a\} & \text{if } x \prec^* a \\ \{x\} & \text{otherwise.} \end{cases} \tag{2}$$

Generalization to a level The generalization to a level transformation is parametrized by a level of generality $n \in \mathbb{N}$. The level n_x is defined inductively for all $x \in F$ as follows. Let $n_x = 0$ for the most general feature x. Let $a, b \in F$. Then $n_b = n_a + 1$ for all b such that $b \prec a$. The transformation causes all features $x \in F$ with level of generality $n_x > n$ to be mapped to their generalization $a \in F$ such that $n_a = n$. This transformation is achieved by the mapping

$$L_n(X) = \bigcup_{x \in X} l_n(x)\,, \text{ where} \tag{3}$$

$$l_n(x) = \begin{cases} \{a\}\,,\ n_a = n,\ x \prec^* a & \text{if } n_x > n \\ \{x\} & \text{if } n_x \leq n. \end{cases} \tag{4}$$

The transformation is closely related to the *heights of generalization* concept introduced by Scott and Matwin (1998). Note that every L_n-transformation can be performed as a composition of G_a-transformations for all features a such that $n_a = n$. However, the expression power of the G_a-transformation is bigger than that of the L_n-transformation.

Omitting a feature The transformation causes the feature a, which is the parameter of this transformation, to be omitted from the feature multiset, including all specializations of a:

$$O_a(X) = \bigcup_{x \in X} o_a(x)\,, \text{ where} \tag{5}$$

$$o_a(x) = \begin{cases} \emptyset & \text{if } x \in \mathcal{S}(a) \bigcup \{a\} \\ \{x\} & \text{otherwise.} \end{cases} \tag{6}$$

The use of this transformation is related to the "wrapper" approach to feature selection (John et al., 1994), where a set of relevant features is chosen iteratively, by greedily adding or removing a single feature until significant decrease in the classification performance is observed. This greedy algorithm yields a locally minimal set of features that maintain the classification performance of the full set of features.

2.3 The algorithm

We apply here a greedy approach to search for the locally optimal transformation. The algorithm assumes the existence of a target function $E\colon \mathcal{M} \to \mathbb{R}$,

where $\mathcal{M}$ is the set of all possible feature mappings. The function E evaluates the goodness of feature mappings $M \in \mathcal{M}$ and can be used to compare two mappings with respect to a criteria represented by the function E. Let further $\Lambda \subseteq \mathcal{M}$ be the set of all defined elementary transformations with all possible parameter combinations, let $\theta \in \mathbb{R}$ be a small threshold constant, and let $I \in \mathcal{M}$ be the identity mapping. The greedy algorithm that returns a locally optimal transformation is presented in Figure 1.

input: Λ, E
output: a mapping $M \in \mathcal{M}$
$i \leftarrow 0$, $M_0 \leftarrow I$, $\Gamma \leftarrow \Lambda$
while $|\Gamma| > 0$
    $i \leftarrow i + 1$
    $M_i^\star \leftarrow \arg\max_{M \in \Gamma} E(M \circ M_{i-1})$
    $M_i \leftarrow M_i^\star \circ M_{i-1}$
    $\Gamma \leftarrow \Gamma \setminus \{M_i^\star\}$
    if $E(M_i) - E(M_{i-1}) < \theta$ **then**
        return M_{i-1}
    end if
end while
return M_i

Fig. 1. A greedy algorithm to compute a locally optimal transformation as the composition of several elementary transformations drawn from the set Λ.

Example 1. Let us consider the task introduced in Section 1, i.e., classification between documents about eucaryotes and procaryotes. Let F be the terms of the MeSH hierarchy and E be a function that evaluates how well a feature mapping M helps some classifier to separate the two classes. The set Λ contains all defined elementary transformations, that is, Λ contains all transformations that generalize up to a MeSH term $\bigcup_{x \in F} G_x$, all transformations that omit a MeSH term $\bigcup_{x \in F} O_x$, and all transformations that generalize to a level $\bigcup_{n=1}^{N} L_n$ where N is the depth of the MeSH hierarchy tree.

The set Λ contains, among others, also the transformations $G_{Organisms}$, $G_{Animals}$, and $G_{Bacteria}$. The transformation $G_{Organisms}$ is obviously harmful, as it suppresses the distinction between eucaryotes and procaryotes. The other two transformations are probably beneficial for any classifier, given the eucaryote vs. procaryote classification problem, since there is no need to distinguish between individual members of the *Bacteria* or *Animal* groups: all animals are eucaryotes and all bacteria are procaryotes. The transformation could, for example, be $G_{Animals} \circ G_{Bacteria} \circ G_{Plants} \circ G_{Fungi} \circ \ldots$ resulting in combining all the various direct specializations of $Organisms$, but never combining all the organisms.

The transformation in this example will affect the classification in at least two ways. Every feature that is a member of, for example, the *Animals* subtree

is replaced with the feature *Animals*. Considering, for example, the maximum-likelihood probability estimate of the naive Bayes classifier, the replacement alleviates the data sparseness problem, because the classifier no longer needs to estimate the class-wise probabilities separately for every individual *Animals* feature. Further, when a document instance is being classified and its feature multiset contains some animal feature, but the particular animal was not encountered during the training of the classifier, the unknown feature can be used in the classification, because due to the transformation $G_{Animals}$ it "inherits" the class-wise characteristics of the feature *Animals*.

2.4 Evaluation function E

The greedy algorithm introduced in Section 2.3 assumes an evaluation function E which can be used to evaluate how well a mapping fulfills the criteria represented by the function E. Here we define the function E in terms of cross-validated classification performance of a classifier using the mapped features on some text classification problem.

Let R be a set of labeled training examples, and let $r \in R$ be a training example. Let further $X_r \subseteq F$ be a multiset of features associated with the example r. Let us assume a classifier $\mathcal{C}\colon 2^F \to \mathbb{N}$ that assigns a class label to the instance r, based on its associated feature multiset X_r. Then, for each feature transformation mapping M, we can define $E(M)$ to be, for example, the average accuracy of $\mathcal{C}$ when performing a 10-fold cross-validation experiment using the set R. For each instance r and its associated feature multiset X_r, the class is computed as $\mathcal{C}(M(X_r))$.

3 Application to a document classification problem

We apply the method to the problem of classifying MEDLINE-indexed documents. The set of transformations Λ is thus instantiated on the MeSH ontology. The evaluation function E is defined in terms of the naive Bayes classifier.

3.1 The MeSH ontology

The MeSH (Medical Subject Headings) is the National Library of Medicine's (NLM) controlled vocabulary of medical and biological terms. MeSH terms are organized in a hierarchy that contains the most general terms (such as *Chemicals and Drugs*) at the top and the most specific terms (such as *Aspirin*) at the bottom. There are 21,973 main headings, termed *descriptors*, in the MeSH.

Publications in the MEDLINE database are manually indexed by NLM using MeSH terms, with typically 10–12 descriptors assigned to each publication. Hence, the MeSH annotation defines for each publication a highly descriptive set of features. Of the over 7 million MEDLINE publications that contain abstracts, more than 96% are currently indexed.

3.2 Feature extraction from MeSH-annotated MEDLINE documents

An occurrence of a term in the MeSH hierarchy is not unique: a term may appear more than once in the hierarchy, as a member of different subtrees. For example the term *Neurons* appears in the subtrees *Cells* and *Nervous system*. The MEDLINE documents are annotated using MeSH terms, rather than their unique subtree numbers, and thus it is not possible to distinguish between the possible instances of the term in the MeSH hierarchy. We separate the ambiguous term occurrences by renaming them, for example, to *Neurons1* and *Neurons2*. When extracting the features (MeSH terms) of a MEDLINE document, we include all possible instances of the ambiguous term occurrence. Thus, a document annotated with the MeSH term *Neurons* will be represented as having two features: *Neurons1* and *Neurons2*. In the following, we will consider the MeSH hierarchy where all ambiguous occurrences of terms have been renamed and thus a term occurrence in this modified MeSH hierarchy is unique. The modified MeSH hierarchy contains 39,853 descriptors. Since the MeSH hierarchy consists of 15 separate trees, we also introduce a single root for the hierarchy.

3.3 Experimental setup

In the empirical evaluation of the method, we consider the following classification problem. We randomly select 10 journals that have at least 2000 documents indexed in the MEDLINE database. For each of these 10 journals, 2000 random documents were retrieved from MEDLINE. The classification task is to assign a document to the correct journal, that is, to the journal in which the document was published. The 10 journals form 10 classification datasets, each having 2000 positive examples and 18000 negative examples formed by the documents belonging to the other 9 journals. The proportion of positive and negative examples is thus 1:9 in each of the datasets.

From the possible elementary transformations presented in Section 2.2, we only consider the generalization to a feature presented in Section 2.2, since the transformation that omits a feature is closely related to a standard and well researched feature-selection technique. The generalization up to a level was tested in our early experiments, but it proved out to be clearly less effective than the generalization to a feature transformation. This is in agreement with the findings of Scott and Matwin (1998). However, note that the MeSH hierarchy requires 10 L-transformations only, whereas up to 11,335 G-transformations need to be evaluated in each step of the greedy algorithm.[6] Adopting the generalization to a feature transformations thus increases the computational requirements significantly.

We use the area under the precision-recall curve (AUC) induced by a leave-one-out cross-validation experiment using the naive Bayes classifier as the value of the evaluation function E. The area under the precision-recall curve is the

[6] The modified MeSH tree has depth 10 and 11,335 non-leaf nodes.

average precision over the whole recall range. The AUC is directly related to the well-known 11-point average precision (see, e.g., Witten and Frank (2000)). To avoid the variance at the extremities of the curve, we use a trimmed AUC only considering the area from 10% recall to 90% recall. We construct the curve by ordering the classified documents in descending order by the positive vs. negative class probability ratio of the naive Bayes classifier and computing the precision and recall values at each of the documents. An important property of the naive Bayes classifier is that it allows implementation of an $O(n)$ complexity leave-one-out cross-validation. A fast implementation of the leave-one-out cross-validation is necessary, since it is performed in each round of the greedy algorithm for each possible elementary transformation. We chose the leave-one-out cross-validation scheme to ensure high stability of the function E, avoiding the variance induced by the random dataset split in, for example, 10-fold cross-validation. Since most of the individual elementary transformations have only a very small effect on the performance, it is important to obtain an accurate and stable measure of the performance in order to distinguish even small gain from noise. The stopping criteria parameter θ is set $\theta = 10^{-4}$.

We cross-validate the results for each of the 10 journal datasets separately, using the 5×2cv cross-validation methodology introduced by Dietterich (1998). The 5×2cv test performs five replications of a 2-fold cross-validation. In each fold, we use the training set data to build the transformation mapping, as described in Section 2.4, and then, using the transformation mapping and the training set data, we estimate the performance of the classifier on the test set data. The test set data is not used during the search for the mapping nor during the training of the classifier. Each of the 5 cross-validation replications consists of two folds. For each fold we measure the standard untrimmed AUC, unlike in the case of the function E, and then average the AUC of the two folds. The performance of the 5 replications is then averaged to obtain the final cross-validated measure of the classification performance for one dataset. To test for statistical significance, we use the robust 5x2cv test (Alpaydm, 1999), since the standard t-test would give misleading results due to the dependency problem of cross-validation.

As the baseline, we use the naive Bayes classifier with no feature transformation applied. In both cases the method introduced by Ng (1997) was used to smooth the maximum-likelihood estimate of the probabilities for the naive Bayes classifier. The Ng's method is commonly applied in text classification tasks and it does not interfere with the $O(n)$ implementation of leave-one-out cross-validation for the naive Bayes classifier.

3.4 Empirical results

The results for the 10 datasets are presented in Table 1.

For 6 datasets, the transformed feature hierarchy results in a statistically significant ($p < 0.05$) increase of the classification performance. Note that for two of the other datasets (datasets 2 and 5) the baseline performance is very close to 100% leaving little room for significant improvement. For the dataset 2, the

#	Journal ID	AUC [%]		Δ [%]	p	rnds	TS [%]
		Baseline	Transf.				
1	ActaAnatBasel	87.15	88.05	0.90	**0.043**	9.0	76.5
2	ApplEnvironMicrobiol	98.28	98.26	-0.02	0.535	0.2	99.7
3	BiolPsychiatry	95.14	95.70	0.56	**0.001**	5.0	80.3
4	EurJObstetGynecol.	91.21	92.31	1.10	**0.006**	8.7	73.0
5	FedRegist	99.48	99.48	0.00	undef.	0.0	100.0
6	JPathol	81.71	82.94	1.23	**0.003**	13.3	84.2
7	NipponRinsho	65.41	67.24	1.83	**0.017**	30.4	75.6
8	PresseMed	51.06	51.38	0.32	0.503	31.4	79.3
9	SchweizRundschMedPrax	58.95	61.53	2.58	**0.029**	25.8	68.4
10	ToxicolLett	88.93	89.12	0.19	0.403	5.5	92.0

Table 1. The classification performance of the naive Bayes classifier. First, the untrimmed AUC percentages are given for the baseline and transformed features. The column denoted Δ is the improvement over the baseline. The column p is the p-value of the 5×2cv statistical significance test. Statistical significance ($p < 0.05$) is denoted in bold face. The column *rnds* is the average number of elementary transformations applied by the greedy algorithm, and the column TS is the average size of the transformed MeSH tree as a percentage of the original size of 39,853 nodes (see Section 3.4 for discussion).

transformed feature hierarchy results in a negligible decrease of the classification performance.

Depending primarily on the number of transformations taken, the processing time varies from 3 minutes (no transformations taken) to 1 hour 45 minutes (44 transformations taken) for each fold, using a 2.8GHz processor.

The G-transformation used in the experimental evaluation can also be considered in terms of dimensionality reduction, since a G_a-transformation causes all features in $\mathcal{S}(a)$ to be replaced with a, hence the classifier never encounters any feature $f \in \mathcal{S}(a)$. The column TS of Table 1 presents the size of the tree when the features f are considered as removed. A reduction to about 80% of the tree size can be typically observed.

4 Discussion

To study the effect of dataset size on the performance of the method, we repeated the experiment for several smaller datasets. We observed, however, no systematic behavior of the 10 datasets with respect to dataset size.

Figure 2 demonstrates a rather good outcome of a classification experiment[7] for a single dataset, where the precision of the classifier with the transformed features is higher than the baseline over the whole recall range. In the experiments, however, it was often the case that the two curves crossed in at least one point, meaning that the transformed features increase the precision only on

[7] The curves represent a real experiment, however.

certain intervals of the recall range, while on other intervals the precision decreases. In such a case, the AUC values of the two curves are roughly similar, yielding a more conservative estimate of the performance than the accuracy at any single point on the curve. A full evaluation of the performance of the method for a single dataset thus ideally requires an analysis of the full precision-recall characteristic of the classifier.

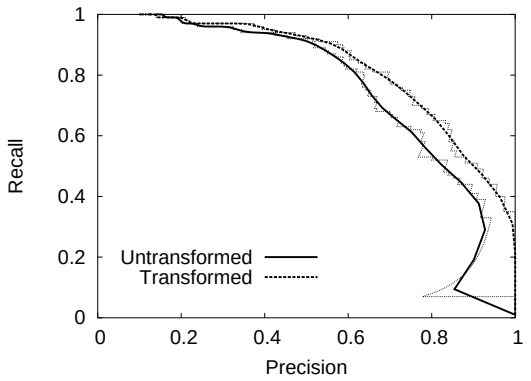

Fig. 2. An example of precision-recall curves of a classifier with transformed and untransformed features. The data is smoothed with a Bézier curve.

The proposed method assumes that the features are organized in some hierarchical ontology. This, however, is not a prohibitive restriction, since any natural language text can be mapped to a general ontology such as WordNet. There also exists a biomedical ontology, the UMLS, which is provided with a tool for mapping unannotated text to the ontology, the MetaMap program. Such a mapping involves several issues, such as ambiguities in the text or the ontology, that can introduce errors into the feature extraction process. A further evaluation of the method on classification tasks involving features obtained by automatic mapping of a free text to an ontology is thus necessary.

The G-transformation used in the empirical evaluation of the method can be viewed as a binary semantic similarity measure, that is, two words can be either synonymous or semantically unrelated. This binary approach can be seen as a special case of a weighted approach, where the weights expressing the strength of a relationship between two words are 1 or 0. Future research can thus be directed to devise methods that compute finer-grained similarity representation tailored to the problem at hand.

5 Conclusions

In this paper we present a novel approach to incorporate semantic information to the problems of natural language processing, in particular to the document

classification task. We devise a theoretical framework in which the semantic information is incorporated in the form of ontology-based feature transformations. We introduce several elementary transformations and an algorithm that identifies a beneficial transformation as a composition of elementary transformations. In order to obtain a feature transformation that is optimized both for the data and the classification method used, we define an evaluation function E that directs the greedy search in terms of the same classification method that is applied to the classification task. This is analogous to the wrapper approach of John et al. (1994).

To test the method empirically, we apply it to a classification problem on MeSH-annotated documents. The empirical results show that the method is capable of statistically significant improvement of performance in 6 out of 10 datasets. In two datasets the improvement was not statistically significant, and for the remaining two datasets no significant improvement can be expected due to the very high baseline performance.

While the results indicate that the presented greedy algorithm is sufficient to validate the concept of feature transformations, it must repeatedly evaluate a potentially large number of elementary transformations, which makes it computationally expensive relative to the baseline method. Further research should thus be directed to devise better search strategies that result in a more efficient algorithm. For example, the search space could be reduced by exploiting the fact that the features are organized in a hierarchy. The search should also attempt to avoid stopping in local optima. Various forms of elementary transformations and evaluation functions E can also be studied. The results show that some datasets benefit more from the method than others. Further work should therefore be directed to study the properties of the data that determine whether a beneficial transformation can be found and how big an improvement can be achieved for the given dataset.

6 Acknowledgments

This work has been supported by Tekes, the Finnish National Technology Agency.

References

Rada, R., Bicknell, E.: Ranking documents with a thesaurus. Journal of the American Society for Information Science **40** (1989) 304–310

Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. In Mellish, C., ed.: Proceedings of the 14th International Joint Conference on Artificial Intelligence, Morgan Kaufmann, San Francisco (1995) 448–453

Budanitsky, A.: Lexical semantic relatedness and its application in natural language processing. Technical Report CSRG390, University of Toronto (1999)

Baker, D., McCallum, A.: Distributional clustering of words for text classification. In Croft, W.B., Moffat, A., van Rijsbergen, C.J., Wilkinson, R., Zobel, J., eds.:

Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM Press, New York (1998) 96–103
Scott, S., Matwin, S.: Text classification using WordNet hypernyms. In Harabagiu, S., ed.: Use of WordNet in Natural Language Processing Systems: Proceedings of the Conference. Association for Computational Linguistics, Somerset, New Jersey (1998) 38–44
John, G.H., Kohavi, R., Pfleger, K.: Irrelevant features and the subset selection problem. In Cohen, W.W., Hirsh, H., eds.: Proceedings of the 11th International Conference on Machine Learning, Morgan Kaufmann, San Francisco (1994) 121–129
Witten, I.H., Frank E.: Data Mining. Morgan Kauffman, San Francisco (2000)
Dietterich, T.G.: Approximate statistical test for comparing supervised classification learning algorithms. Neural Computation **10** (1998) 1895–1923
Alpaydm, E.: Combined 5×2 cv F test for comparing supervised classification learning algorithms. Neural Computation **11** (1999) 1885–1892
Ng, H.T.: Exemplar-based word sense disambiguation: Some recent improvements. In Cardie, C., Weischedel, R., eds.: Proceedings of the Second Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Somerset, New Jersey (1997) 208–213