Document Classification Using Semantic Networks with An Adaptive Similarity Measure

Filip Ginter, Sampo Pyysalo, and **Tapio Salakoski**
Turku Centre for Computer Science (TUCS) and
Department of IT, University of Turku
Lemminkäisenkatu 14 A
Turku 20520, Finland
firstname.lastname@it.utu.fi

Abstract

We consider supervised document classification where a semantic network is used to augment document features with their hypernyms. A novel document representation is introduced in which the contribution of the hypernyms to document similarity is determined by semantic network edge weights. We argue that the optimal edge weights are not a static property of the semantic network, but should rather be adapted to the given classification task. To determine the optimal weights, we introduce an efficient gradient descent method driven by the misclassifications of the k-nearest neighbor (kNN) classifier. The method iteratively adjusts the weights, increasing or decreasing the similarity of documents depending on their classes.

We thoroughly evaluate the method using ten randomly chosen datasets and seven training set sizes on the problem of classifying PubMed documents indexed with the MeSH biomedical ontology. Using the kNN classifier, the method is shown to statistically significantly outperform the commonly used bag-of-words representation as well as the more advanced hypernym density representation (Scott & Matwin 98).

1 Introduction

Semantic networks have been shown to offer opportunities for improving the performance of both supervised and unsupervised machine learning methods in a variety of classification tasks. Several semantic similarity measures have been proposed and applied in particular to word sense disambiguation-type problems, where the similarity between each ambiguous word candidate and the context words can be used to choose between the candidates (see e.g. (Budanitsky & Hirst 01; Patwardhan *et al.* 03) for recent evaluations). Methods applying semantic networks to document classification, where the class labels are not themselves part of the semantic network have also been proposed, although they are not as widely studied. For instance, semantic networks have been used to augment the terms occurring in documents with their synonyms (Gomez-Hidalgo & deBuenaga Rodriguez 97) and hypernyms (Scott & Matwin 98; Bloehdorn & Hotho 04), thus incorporating the information encoded in semantic networks on the level of features. Semantic networks have also been applied in modeling document similarity for a kernel-based document classification method (Basili *et al.* 05).

The similarity of terms is typically presented as a static property that can be directly measured either from the semantic network (Leacock & Chodorow 98; Agirre & Rigau 96), from external (unlabeled) data (Resnik 95), or using a combination of the two (Jiang & Conrath 97). In this paper, we consider the special case of similarity through the hyponymy/hypernymy relation, which is the focus of most proposed measures of semantic relatedness.

We have previously argued that in supervised classification tasks the similarity of terms should be considered dependent on the task and data (Ginter *et al.* 04). Simply put, terms commonly related to documents of the same class should be considered similar, while terms related to documents of different classes should be considered dissimilar to aid the classification method in distinguishing between the classes.

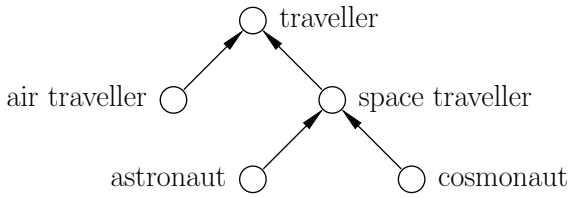
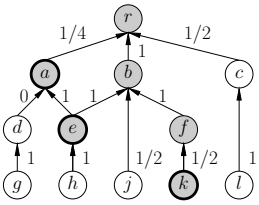


Figure 1: *Hyponymy.* The arrows represent hyponymy relationships between terms in a fragment of a semantic network.

To illustrate this idea, consider the fragment of a semantic network shown in Figure 1. Common measures of semantic similarity would assign high relatedness to the terms *astronaut* and *cosmonaut* as they are immediate hyponyms of the same term, *space traveller*. In most classification tasks, considering *astronaut* and *cosmonaut* essentially synonymous terms would be appropriate. In a document representation, this can be naturally realized by considering the term *space traveller* to be highly relevant to documents containing either of its two hyponyms. However, we suggest that in a hypothetical document classification task where the goal is to distinguish between documents about American and Russian space efforts, *space traveller* should not be considered relevant to documents containing either *astronaut* or *cosmonaut* to avoid increasing the similarity between documents of different classes.

We now discuss some desirable properties for a data-dependent semantic document representation and



$Aff(d) = \{e,\, a,\, k,\, b,\, r,\, f\}$

$Base_r(d) = \{a,\, b\}\,,\ Base_a(d) = \{e\}\,,\ Base_b(d) = \{e,\, f\}\,,\ Base_f(d) = \{k\}$

$Base_g(d) = Base_h(d) = Base_j(d) = Base_k(d) = Base_l(d) = Base_d(d) = Base_e(d) = Base_c(d) = \emptyset$

$a_e(d) = a_a(d) = a_k(d) = 1,\ a_d(d) = a_g(d) = a_h(d) = a_j(d) = a_l(d) = a_c(d) = 0$

$$a_f(d) = w_{kf} a_k(d) = \frac{1}{2},\ a_b(d) = \frac{w_{eb} a_e(d) + w_{fb} a_f(d)}{2} = \frac{3}{4},\ a_r(d) = \frac{w_{ar} a_a(d) + w_{br} a_b(d)}{2} = \frac{1}{2}$$

Figure 2: *Document representation.* An example illustrating the representation of a document d with direct terms $T(d) = \{e,\, a,\, k\}$. Direct terms of d are denoted by bold circles and terms affected by d are depicted in gray. The semantic network weights are shown by each edge.

means of realizing them. We assume that each document has been assigned a set of *direct* terms from a semantic network (e.g. terms that are mentioned in the document, or relevant keywords that have been assigned to the document). The representation should then determine the relevance of each semantic network term for each document. It is natural to limit this measure of relevance between 0 and 1, and to assign the value 1 to each direct term. As illustrated by the *astronaut/cosmonaut* example, hypernyms of direct document terms are typically relevant and their relevance values should be allowed to vary in a data-dependent fashion. We suggest that terms that are neither direct terms nor hypernyms of direct terms in a document are not relevant to that document and can be assigned the relevance value 0: for example, if *astronaut* is the only direct term, there is no reason to assume that either *cosmonaut* or *air traveller* are relevant. Finally, relevance should not increase with distance from the direct term: if, for example, *astronaut* is the only direct term, *traveller* should be considered at most as relevant as *space traveller*. This implies a representation where relevance propagates from direct terms to more general terms, decreasing according to the data-dependent strengths of connections between hyponyms and hypernyms.

We have previously introduced a data-driven method for determining hypernym relevance for document classification (Ginter *et al.* 04), where relevance was limited to the two cases "fully relevant" and "irrelevant", achieving a modest yet statistically significant 0.9 percentage unit average performance increase from a 81.7% bag-of-words baseline (average precision measure). In this paper, we present a method that applies a finer-grained concept of relevance and shows a more substantial performance advantage.

2 Document representation

Let $\mathcal{T}$ be a finite set of possible terms that are organized in a semantic network according to the semantic relation of hyponymy. Let $t, t' \in \mathcal{T}$ be terms. We denote by $t' \prec^\star t$ the relation when t' is a hyponym of t. Further, $t' \prec t$ denotes the relation when t' is an *immediate hyponym* of t, that is, the relation encoded by the semantic network. Hyponymy ($\prec^\star$) is the transitive closure of immediate hyponymy ($\prec$). For example, we have *astronaut* $\prec$ *space traveller*, *astronaut* $\prec^\star$ *traveller*, but *astronaut* $\not\prec$ *traveller*. The immediate hyponymy relation between the terms in $\mathcal{T}$ is commonly represented as a directed graph, such as the graph in Figure 1, with an edge from t' to t whenever $t' \prec t$. Hyponymy ($\prec^\star$) is by definition an asymmetric relation, and the corresponding directed graph is thus acyclic.

We define a document representation that implements the intuitions discussed in Section 1. Let $\mathcal{D}$ be a set of documents and let $d \in \mathcal{D}$ be a document with the set of direct terms $T(d) \subseteq \mathcal{T}$. As discussed previously, the document d is represented not only by the direct terms in $T(d)$ but also by their hypernyms. The proposed document representation implements this property through the notion of *activation* $a_t(d) \in [0, 1]$ of a term $t \in \mathcal{T}$ with respect to the document d that represents the relevance of t to d. For any term t that belongs to $T(d)$, $a_t(d)$ is by definition set to 1, the maximum possible activation value. The activation of any other term recursively depends on the activations of its immediate hyponyms so that the activation of hypernyms of direct terms typically results in a non-zero value. The activation of the remaining terms is zero by definition.

We say that a term $t \in \mathcal{T}$ is *affected* by a document d if $t \in T(d)$ or $\exists\, t' \in T(d) : t' \prec^\star t$. That is, t is affected by d if t is either a direct term of d or a hypernym of a direct term. The set of all terms affected by a document d is denoted $Aff(d)$. Let us further define the *base* of a term $t \in \mathcal{T}$ with respect to a document d as the set of immediate hyponyms of t that are affected by d. Formally,

$$Base_t(d) = \{t' \,|\, t' \prec t,\, t' \in Aff(d)\}\ .$$

Unless t is a direct term, its activation is based on the activations of the terms in $Base_t(d)$. For each term t' in the base of t, the contribution of t' to the activation of t is controlled by a *weight* $w_{t't}$ that is associated with the relationship $t' \prec t$. By definition, $0 \leq w_{t't} \leq 1$ for all weights in the semantic network. The activation $a_t(d)$ is computed as the weighted sum

of the activations of the terms in $Base_t(d)$. Thus,

$$a_t(d) = \begin{cases} 1 & \text{if } t \in T(d), \\ \frac{\sum_{t' \in Base_t(d)} w_{t't} a_{t'}(d)}{|Base_t(d)|} & \text{if } t \in Aff(d) \setminus T(d), \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Each document d is then represented in classification by its *activation vector* $a(d)$,

$$a(d) = (a_{t_1}(d), \ldots, a_{t_m}(d)) ,$$

where $t_k \in \mathcal{T}$, $1 \leq k \leq m$, and $m = |\mathcal{T}|$. Figure 2 illustrates the concepts introduced so far.

Note the following special cases of the document representation. If all weights in the network are set to 0, the document is represented by the set of its direct terms and the representation is thus equivalent to the common bag-of-words (BoW) representation—here we do not consider the case where duplicate terms occur. If all weights are set to 1, the document is represented by the set of its direct terms together with all their hypernyms, an intuitively plausible representation as well.

3 Weight update algorithm

In this section, we describe an algorithm that optimizes the semantic network weights in order to maximize the classification performance on a given document classification task. The algorithm thus implements the adaptive component of the proposed method. In short, the algorithm initializes all weights to 1 and then iteratively adjusts the weights until no more improvement in classification performance can be achieved. The algorithm implements the gradient-descent search strategy.

3.1 Document similarity and classification

Let $\hat{a}(d)$ be the *normalized activation vector* of d, that is,

$$\hat{a}(d) = \frac{a(d)}{\|a(d)\|} . \quad (2)$$

We calculate the similarity between any two documents $d_i, d_j \in \mathcal{D}$ from their normalized activation vectors with the commonly used dot-product measure

$$\mathrm{sim}(d_i, d_j) = \hat{a}(d_i) \cdot \hat{a}(d_j) = \sum_{t \in \mathcal{T}} \hat{a}_t(d_i) \hat{a}_t(d_j) . \quad (3)$$

The weight update algorithm is based on the k-nearest neighbor (kNN) classifier. Given a training set of documents $\mathcal{D}$, a document similarity measure, and a document d to be classified, the kNN classifier computes a set $N(d, k, \mathcal{D}) \subseteq \mathcal{D}$ of k documents most similar to d, also termed as the *k-neighborhood*. The document d does not itself belong to its k-neighborhood. The class assigned to d is the majority class among the documents in its k-neighborhood.

3.2 Weight update

The weight update algorithm implements the following intuition. As the documents are classified using the kNN classifier, a misclassification of a document d means that the majority of the documents in $N(d, k, \mathcal{D})$ are of a different class than d. The misclassification could therefore be corrected by modifying the k-neighborhood so that it would contain a majority of documents with the same class as that of d. This can be achieved by adjusting the semantic network weights, and thus the document representation, so that the similarity between d and its k-neighbors with a different class decreases and the similarity between d and its k-neighbors with the same class increases. As there is only one, global set of weights, any change affects all the documents and therefore directly optimizing the similarity of d with its k-neighbors also indirectly affects the similarity of d with all other documents. Generally, documents with the same class are "pulled" towards d while documents with another class are "pushed" away from d. Naturally, this effect is strongest for the k-neighbors of d, whose similarity with d is optimized directly. As the other class k-neighbors are "pushed" away from d, they are replaced in the k-neighborhood by same class documents that are "pulled" towards d. Other variations of the general scheme are possible as well. For example, the k-neighborhoods could be optimized for all documents rather than only for those that were misclassified.

Let us consider two documents $d_i, d_j \in \mathcal{D}$. The objective is to either increase or decrease $\mathrm{sim}(d_i, d_j)$ by modifying the semantic network weights. Let us define the vector w of all weights in the semantic network in an arbitrary but fixed order

$$w = (w_1, \ldots, w_n) ,$$

where n is the total number of weights. We then define the weight gradient $\nabla w(d_i, d_j)$ with respect to $\mathrm{sim}(d_i, d_j)$ as

$$\nabla w(d_i, d_j) = \left(\frac{\partial \,\mathrm{sim}(d_i, d_j)}{\partial w_1}, \ldots, \frac{\partial \,\mathrm{sim}(d_i, d_j)}{\partial w_n} \right) .$$

Adding the gradient $\nabla w(d_i, d_j)$ to the weight vector w leads to an increase of $\mathrm{sim}(d_i, d_j)$, while subtracting $\nabla w(d_i, d_j)$ from w leads to a decrease of $\mathrm{sim}(d_i, d_j)$. The formula to compute the partial derivative $\frac{\partial \,\mathrm{sim}(d_i, d_j)}{\partial w_{rs}}$ of $\mathrm{sim}(d_i, d_j)$ with respect to a weight w_{rs} is fully specified jointly by Equations 5, 11, and 12 in Appendix A which also details the derivation leading to the formula.

A *learning rate* constant $\eta \in \mathbb{R}$, $\eta > 0$, is introduced to control the magnitude of the weight adjustment by the gradient. The weight vector w is then updated according to the rule

$$w \leftarrow w + \delta \eta \nabla w(d_i, d_j) ,$$

where $\delta = +1$ (resp. $\delta = -1$) if $\mathrm{sim}(d_i, d_j)$ is to be increased (resp. decreased).

The complete weight update algorithm is introduced in Algorithm 1. In each iteration, the weight adjustments $\delta\nabla w(d_i, d_j)$ are summed into w' over all the document pairs (d_i, d_j) where d_i was misclassified and d_j belongs to its k-neighborhood. Subsequently, w', scaled by the learning rate η, is added to the weight vector w. Finally, each weight w_k in w is clipped such that the constraint $0 \leq w_k \leq 1$ holds. The iteration is finished using some stopping criterion, for example the classification performance failing to increase, which signals that the algorithm has reached a local optimum.

```
w ← 1̄
while not done:
    w' ← 0̄
    for each document d_i ∈ D:
        classify d_i using D \ {d_i} as training set
        if misclassified d_i then:
            for each d_j ∈ N(d_i, k, D \ {d_i}):
                if class(d_i) = class(d_j) then:
                    δ ← +1
                else
                    δ ← −1
                w' ← w' + δ∇w(d_i, d_j)
    w ← w + η · w'
    for each weight w_k in w:
        w_k ← max{0, min{1, w_k}}
```

Algorithm 1: Pseudocode of the weight update algorithm.

3.3 Implementation issues

An efficient implementation of the algorithm can be achieved through the following observation. Let us consider a weight w_{rs} and a term t such that t is not a hypernym of s. From the definition of activation, it is clear that $a_t(d)$ is constant with respect to w_{rs} and thus $\frac{\partial a_t(d)}{\partial w_{rs}} = 0$. Consequently, when computing $\frac{\partial \mathrm{sim}(d_i, d_j)}{\partial w_{rs}}$, it is only necessary to evaluate Equation 12 for s and its hypernyms instead of all terms in $\mathcal{T}$. The computation time of a single partial derivative $\frac{\partial \mathrm{sim}(d_i, d_j)}{\partial w_{rs}}$ is thus constant with respect to $|\mathcal{T}|$. It depends on the number of terms affected by d_i and d_j, which is typically several orders of magnitude smaller than $|\mathcal{T}|$.

Combining this observation and an efficient computation of the partial derivatives based on a linear walk through the semantic network in topological order, we were able to implement the computation of w' with the complexity $O(cM)$ with respect to the training set size M. Roughly, the constant c quadratically depends on the number of terms affected by the documents d_i and d_j and linearly depends on the k-neighborhood size.

4 Evaluation

In this section, we discuss the evaluation datasets, the experimental setup and the baseline methods.

4.1 Datasets

To evaluate the methods, a set of document classification tasks was required where the direct terms of documents belong to a semantic network. We consider datasets consisting of articles from the PubMed biomedical literature database[1], where each article has been manually assigned a set of relevant terms from the MeSH ontology[2]. This approach allows us to evaluate the method using large datasets and the use of manually assigned direct terms to represent the documents avoids the potential sources of error related to automatic mapping to a semantic network.

The datasets were formed as follows: for each dataset, a journal was selected that contains at least 2000 MeSH-indexed articles with abstracts; here we use the 10 journals we selected randomly in (Ginter *et al.* 04). Then, for each of the 10 journals, we randomly selected 2000 articles that have appeared in the journal (as positives) and 2000 that have appeared elsewhere (as negatives). Each task is then a binary classification problem where the documents must be classified either as originating from the journal or not. Since the journals are usually focused on a subdomain, these classification problems model document classification by topic.

To determine the performance of the methods with respect to different training set sizes, we formed for each dataset seven different training sets, the largest consisting of 1000 positive and 1000 negative examples (the other 2000 being used for testing). Smaller training sets were formed by downsampling so that the size is repeatedly halved.

4.2 Methods and performance measurement

We evaluate the proposed document representation with and without the adaptive component. In the *fixed* representation, the semantic network weights are all set to one constant value w_{fix}, $0 \leq w_{fix} \leq 1$, determined from the data. In the *adaptive* representation, the weights are computed using the algorithm introduced in Section 3, using a stopping criterion where iteration ends when the average performance increase on the training set over the last three rounds drops below 0.05%.

We compare the performance of the fixed and adaptive representations against two baselines, the commonly used bag-of-words (BoW) representation and a modification of the hypernym density (HD) representation (Scott & Matwin 98). In the BoW representation, each document is represented by its direct terms.

[1] `http://www.pubmed.com`

[2] We use the 2005 version of MeSH, available at `http://www.nlm.nih.gov/mesh/`

	1				2				3				4				5			
	BoW	HD	Fix.	Ad.	BoW	HD	Fix.	Ad.	BoW	HD	Fix.	Ad.	BoW	HD	Fix.	Ad.	BoW	HD	Fix.	Ad.
31	69.3	74.4	74.4	79.4	81.4	84.0	84.7	86.0	71.6	76.6	71.7	79.2	73.8	74.8	75.9	76.2	97.0	95.5	96.4	95.5
62	71.1	79.5	79.4	85.0	83.0	86.2	87.1	87.8	75.5	77.4	77.3	82.2	75.9	78.1	79.4	82.6	94.8	96.3	96.4	96.8
125	71.2	81.2	81.1	86.3	86.6	88.7	89.0	90.3	77.4	80.8	79.6	86.8	79.8	81.1	81.7	83.7	96.3	96.3	96.7	97.8
250	72.7	83.6	83.2	88.3	88.7	90.3	90.3	90.8	80.4	83.9	83.4	88.4	80.6	83.6	83.8	87.3	97.1	96.6	97.2	98.0
500	74.9	85.2	85.0	89.1	89.1	90.6	90.5	91.7	82.3	85.8	85.1	90.5	83.6	85.8	86.2	88.9	98.0	97.4	98.3	98.2
1000	76.5	86.5	86.3	89.8	90.3	91.8	91.6	92.5	84.0	87.7	87.3	91.0	85.8	87.7	87.2	90.1	97.8	97.5	98.1	98.3
2000	78.7	87.8	88.0	91.1	91.2	92.2	92.1	92.7	86.8	89.3	88.8	91.9	87.1	88.4	88.7	90.5	97.7	97.9	97.9	98.6

	6				7				8				9				10			
	BoW	HD	Fix.	Ad.	BoW	HD	Fix.	Ad.	BoW	HD	Fix.	Ad.	BoW	HD	Fix.	Ad.	BoW	HD	Fix.	Ad.
31	64.6	72.9	70.9	71.0	65.1	64.3	66.2	64.7	65.0	66.0	65.4	66.9	65.8	67.5	67.4	69.1	71.8	71.4	71.2	70.9
62	67.3	74.8	74.2	76.5	64.5	68.0	67.1	67.9	66.3	67.4	66.9	68.9	68.1	71.8	71.1	73.1	72.4	73.3	73.1	75.5
125	70.3	77.6	76.3	79.1	65.8	69.9	70.4	70.8	67.8	69.5	68.3	70.9	69.9	71.6	71.4	74.8	73.9	76.8	76.8	79.6
250	75.2	80.2	80.1	82.8	66.5	75.0	73.5	74.7	69.2	71.5	70.8	71.9	72.1	74.6	74.5	76.2	75.1	78.3	77.7	81.7
500	77.3	81.9	82.0	83.8	68.7	76.9	76.1	77.8	70.1	73.3	72.5	73.5	74.5	75.9	75.3	77.4	77.5	81.1	80.4	83.1
1000	80.9	84.5	84.2	85.5	68.3	78.1	76.9	79.4	71.2	74.3	73.6	75.3	76.2	77.5	77.1	78.7	79.2	83.0	82.6	84.9
2000	82.8	85.7	85.5	86.4	69.3	79.8	77.9	80.9	72.9	75.8	75.4	76.2	77.4	78.3	78.7	79.7	80.7	83.8	83.9	85.7

Table 1: *Classification performance of kNN.* Cross-validated accuracy measurements for each of the ten datasets and each of the seven training set sizes. The MEDLINE abbreviations of the corresponding journal names are, in order, *Acta Anat (Basel), Appl Environ Microbiol, Biol Psychiatry, Eur J Obstet Gynecol Reprod Biol, Fed Regist, J Pathol, Nippon Rinsho, Presse Med, Schweiz Rundsch Med Prax,* and *Toxicol Lett.*

In the HD representation, each document d_i is represented by a multiset consisting of all direct terms of d_i, together with their hypernyms up to a distance h from any of the direct terms. We modified the HD representation as follows. We found that in our case coercing the multiset into a set results in an improvement of performance, and thus we apply this step in our evaluation. Further, infrequent terms are not discarded. The HD normalization step is performed by the classifiers. Note that for $h = 0$ the HD representation is equivalent to the BoW representation, and for $h = \infty$ it is equivalent to the fixed representation with $w_{fix} = 1$.

The main evaluation was performed using the kNN classifier. In this evaluation, the parameters of the various methods—k for BoW, k, h for HD, k, w_{fix} for fixed, and k, η for adaptive—were selected separately in each fold by cross-validated grid search on the training set. To assess the applicability of the representations to other classification methods, we also performed a limited evaluation using Support Vector Machines (SVM), a state-of-the-art machine learning method (Vapnik 98). For this evaluation, only the SVM regularization parameter C was separately selected, while other parameters were set to their kNN optimum values.

We measure the performance of the various methods using average 5×2 cross-validated accuracy, reporting differences in accuracy as well as relative decreases in error rate to better estimate the performance of the methods with respect to different baselines. To assess the statistical significance of results for individual datasets, we use the robust 5×2 cross-validation test (Alpaydin 99). To assess the overall significance across all datasets, we use the standard two-tailed paired t-test.

5 Results and discussion

Results with kNN are given in Table 1, and average differences are plotted in Figure 3. Averages are also given in Table 2a.

As can be seen in Figure 3, the adaptive method statistically significantly outperforms all others for all except the smallest training set size. For training set sizes 62 and larger, the adaptive method outperforms BoW by 5–6 percentage units, reflecting a relative decrease in error rate systematically over 20% and approaching 30% for large dataset sizes. The fixed and HD representations also perform well against BoW, both achieving a statistically significant increase in accuracy of 3–4 percentage units (12–20% relative decrease in error rate) for all but the smallest training set size. The differences between these two representations suggest a small (0.1–0.4 percentage unit) advantage to the HD representation, but this difference is largely not statistically significant. Against the fixed and HD representations, the adaptive method offers an accuracy increase between 1 and 3 percentage units, that is, a systematic relative decrease in error rate of 10–14% for all but the smallest training set size.

Further, the average absolute performance advantage of the adaptive method over the BoW baseline grows with increasing training set size from 31 to 250 examples, and falls thereafter. In contrast, in terms of relative decrease in error rate this performance advantage grows almost monotonically, indicating that the adaptive method works better given more data. As the documents were assigned on average only 10

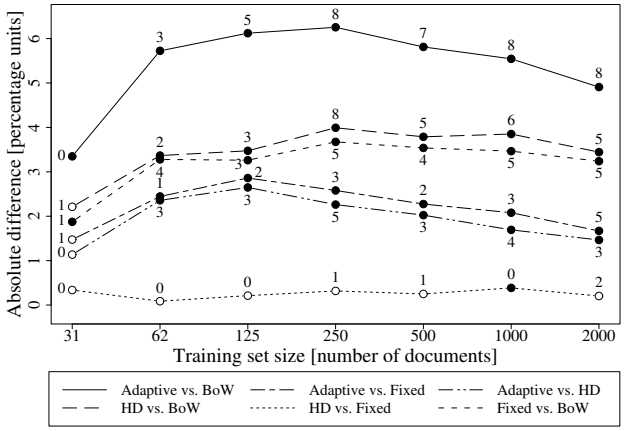


Figure 3: *Pairwise method differences and their per-dataset and overall statistical significances for kNN.* Results averaged over all datasets. The number displayed by each difference denotes the number of individual datasets for which the difference was statistically significant ($p < 0.05$, 5×2cv test). Full circle, as opposed to empty circle, denotes the average difference over all ten datasets being statistically significant ($p < 0.05$, t-test).

MeSH terms and the MeSH ontology contains almost 23000 nodes, reliable optimization of the edge weights is expected to be difficult with very small training sets. Nevertheless, the adaptive method works remarkably well with as few as 62 training examples.

	BoW	HD	Fix.	Ad.
31	72.5	74.8	74.4	75.9
62	73.9	77.3	77.2	79.6
125	75.9	79.4	79.1	82.0
250	77.8	81.8	81.4	84.0
500	79.6	83.4	83.1	85.4
1000	81.0	84.9	84.5	86.6
2000	82.4	85.9	85.7	87.4

(a)

	BoW	HD	Fix.	Ad.
31	75.6	79.1	78.6	78.1
62	78.2	81.6	81.5	80.9
125	80.9	84.0	84.3	83.0
250	83.2	85.8	85.9	84.7
500	85.4	87.5	87.8	86.7
1000	87.5	88.9	89.2	88.4
2000	88.9	90.0	90.3	89.7

(b)

Table 2: *Classification performance.* Accuracy measurements averaged over all ten datasets for each of the seven training set sizes: (a) kNN results, (b) SVM results.

We now present the results of the evaluation with SVM. The average SVM results over the ten datasets are given in Table 2b. The BoW baseline is again outperformed by the other three representations for all training set sizes, with relative decrease in error rate ranging between 12–18% for the fixed representation, 10–17% for the HD representation, and 6–13% for the adaptive method. These differences are statistically significant for all training set sizes for the fixed and HD representations and for training set sizes of 500 and larger for the adaptive method.

We observe that when applied to SVMs, the fixed and HD representations outperform the adaptive method. The difference is statistically significant for most training set sizes larger than 62, where the relative decrease in error rate over the adaptive method ranges between 2–9%. The SVM classification principle substantially differs from that of kNN. Clearly, the adaptive method does not optimize a criterion beneficial for SVM classification, and hence modification of the adaptive strategy is required to increase applicability to SVM classification. Nevertheless, as the fixed representation outperforms both the BoW and HD representations for larger training set sizes (the latter difference is mostly not statistically significant), the general strategy appears to apply well also to SVM.

6 Conclusions and future work

In this paper, we have developed the idea that semantic networks can be used to develop an adaptive document similarity measure. We have discussed desirable properties for such a measure and presented a document representation that implements these properties. Further, we have introduced a gradient descent-based algorithm driven by misclassifications that adapts the representation to data. We have evaluated the representation and the algorithm against the BoW, fixed and HD representations with ten randomly selected datasets from the PubMed biomedical literature database using MeSH 2005 terms as features.

Our results indicate that the proposed adaptive method can statistically significantly outperform the commonly used BoW representation and the more advanced HD representation as well as our new representation with fixed weights over a range of training set sizes from 62 to 2000, for which the relative decrease in error rate ranged between 20–30% against BoW and 10–14% against the fixed and HD representations.

A separate evaluation with Support Vector Machines indicated that while the semantic network-based document representations give a statistically significant improvement over the BoW baseline and the proposed representation performs as well as the HD representation, the gradient descent component of the adaptive method, driven by kNN misclassifications, requires modification to apply beneficially to SVMs. A possible future direction would thus be to introduce the gradient descent algorithm into the SVM training phase, potentially leading to further performance improvements for the classifier.

We conclude that the proposed adaptive similarity measure can successfully determine term-document relevance in a data-dependent manner, increasing performance in supervised document classification tasks. As future work, several aspects of the proposed method can be studied, such as the setting of the initial weights, the learning rate, and the stopping criterion. An additional natural extension of the method is to consider relationships other than hyponymy as activation paths. Careful analysis of these and other properties may offer further opportunities for the use of semantic networks in document classification.

Acknowledgments

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

References

(Agirre & Rigau 96) E. Agirre and G. Rigau. Word sense disambiguation using conceptual density. In *Proceedings of the 16th conference on Computational linguistics COLING '96, Copenhagen, Denmark*, pages 16–22, 1996.

(Alpaydin 99) E. Alpaydin. Combined 5×2 cv F-test for comparing supervised classification learning algorithms. *Neural Computation*, 11(8):1885–1892, 1999.

(Basili *et al.* 05) R. Basili, M. Cammisa, and A. Moschitti. Effective use of WordNet semantics via kernel-based learning. In I. Dagan and D. Gildea, editors, *Proceedings of the Ninth Conference on Computational Natural Language Learning CoNLL 2005, Ann Arbor, Michigan*, pages 1–8. Association for Computational Linguistics, 2005.

(Bloehdorn & Hotho 04) S. Bloehdorn and A. Hotho. Boosting for text classification with semantic features. In *Proceedings of the MSW 2004 workshop, 10th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Seattle*, pages 70–87, 2004.

(Budanitsky & Hirst 01) A. Budanitsky and G. Hirst. Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures. In *Workshop on WordNet and Other Lexical Resources, Second meeting of the North American Chapter of the Association for Computational Linguistics*, pages 29–34, Pittsburgh, 2001.

(Ginter *et al.* 04) F. Ginter, S. Pyysalo, J. Boberg, J. Järvinen, and T. Salakoski. Ontology-based feature transformations: A data-driven approach. In J. L. Vicedo, P. Martínez-Barco, R. Muñoz, and M. Saiz Noeda, editors, *Proceedings of the 4th International Conference EsTAL 2004*, pages 279–290. Springer, Heidelberg, 2004.

(Gomez-Hidalgo & deBuenaga Rodriguez 97) J. M. Gomez-Hidalgo and M. de Buenaga Rodriguez. Integrating a lexical database and a training collection for text categorization. In *Proceedings of the ACL/EACL 97 Workshop on Automatic Information Extraction and Building of Lexical Semantic Resources, Madrid, Spain*, pages 39–44, 1997.

(Jiang & Conrath 97) J. Jiang and D. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the International Conference on Research in Computational Linguistics*, pages 19–33. Academica Sinica, 1997.

(Leacock & Chodorow 98) C. Leacock and M. Chodorow. Combining local context and wordnet similarity for word sense identification. In C. Fellbaum, editor, *WordNet: An Electronic Lexical Database*, pages 265–283. MIT Press, Cambridge, MA, 1998.

(Patwardhan *et al.* 03) S. Patwardhan, S. Banerjee, and T. Pedersen. Using measures of semantic relatedness for word sense disambiguation. In A. Gelbukh, editor, *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*, pages 241–257, Mexico City, Mexico, 2003. Springer-Verlag, Heidelberg.

(Resnik 95) P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In C. Mellish, editor, *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 448–453. Morgan Kaufmann, San Francisco, 1995.

(Scott & Matwin 98) S. Scott and S. Matwin. Text classification using WordNet hypernyms. In S. Harabagiu, editor, *Proceedings of Use of WordNet in Natural Language Processing Systems*, pages 38–44, Somerset, New Jersey, 1998. Association for Computational Linguistics.

(Vapnik 98) V. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

A Derivation of the formula for $\frac{\partial \operatorname{sim}(d_i, d_j)}{\partial w_{rs}}$

This appendix details the derivation of the formula to compute the value of $\frac{\partial \operatorname{sim}(d_i, d_j)}{\partial w_{rs}}$. Starting from (4), the partial derivative is solved and the final formula is obtained jointly from Equations 5, 11, and 12.

$$\frac{\partial \operatorname{sim}(d_i,\, d_j)}{\partial w_{rs}} = \frac{\partial \hat{a}(d_i) \cdot \hat{a}(d_j)}{\partial w_{rs}} = \sum_{t \in \mathcal{T}} \frac{\partial \hat{a}_t(d_i)\hat{a}_t(d_j)}{\partial w_{rs}} = \sum_{t \in \mathcal{T}} \frac{\partial \hat{a}_t(d_i)}{\partial w_{rs}} \hat{a}_t(d_j) + \sum_{t \in \mathcal{T}} \frac{\partial \hat{a}_t(d_j)}{\partial w_{rs}} \hat{a}_t(d_i) \quad (4)$$

Let

$$\frac{\partial \operatorname{sim}(d_i,\, d_j)}{\partial w_{rs}} \stackrel{def}{=} Q(d_i,\, d_j) + Q(d_j,\, d_i)\,, \quad (5)$$

where

$$Q(d_i,\, d_j) \stackrel{def}{=} \sum_{t \in \mathcal{T}} \frac{\partial \hat{a}_t(d_i)}{\partial w_{rs}} \hat{a}_t(d_j) \quad (6)$$

In the following, we solve $Q(d_i,\, d_j)$; the formula for $Q(d_j,\, d_i)$ follows by symmetry.

$$\frac{\partial \hat{a}_t(d_i)}{\partial w_{rs}} \stackrel{(2)}{=} \frac{\partial \frac{a_t(d_i)}{\|a(d_i)\|}}{\partial w_{rs}} = \frac{\frac{\partial a_t(d_i)}{\partial w_{rs}} \|a(d_i)\| - a_t(d_i) \frac{\partial \|a(d_i)\|}{\partial w_{rs}}}{\|a(d_i)\|^2} \quad (7)$$

$$\frac{\partial \|a(d_i)\|}{\partial w_{rs}} = \frac{\partial \sqrt{\sum_{u \in \mathcal{T}} [a_u(d_i)]^2}}{\partial w_{rs}} = \frac{1}{2\|a(d_i)\|} \frac{\partial \sum_{u \in \mathcal{T}} [a_u(d_i)]^2}{\partial w_{rs}} = \frac{1}{\|a(d_i)\|} \sum_{u \in \mathcal{T}} \left(a_u(d_i) \frac{\partial a_u(d_i)}{\partial w_{rs}} \right) \quad (8)$$

Combining (7) and (8) yields

$$\frac{\partial \hat{a}_t(d_i)}{\partial w_{rs}} = \frac{\frac{\partial a_t(d_i)}{\partial w_{rs}} \|a(d_i)\| - \hat{a}_t(d_i) \sum_{u \in \mathcal{T}} \left(a_u(d_i) \frac{\partial a_u(d_i)}{\partial w_{rs}} \right)}{\|a(d_i)\|^2} \quad (9)$$

Substituting from (9) into (6) gives

$$\begin{aligned} Q(d_i,\, d_j) &= \frac{\sum_{t \in \mathcal{T}} \frac{\partial a_t(d_i)}{\partial w_{rs}} \|a(d_i)\| \hat{a}_t(d_j) - \sum_{t \in \mathcal{T}} \left(\hat{a}_t(d_j) \hat{a}_t(d_i) \sum_{u \in \mathcal{T}} a_u(d_i) \frac{\partial a_u(d_i)}{\partial w_{rs}} \right)}{\|a(d_i)\|^2} \\ &= \frac{\sum_{t \in \mathcal{T}} \frac{\partial a_t(d_i)}{\partial w_{rs}} \|a(d_i)\| \hat{a}_t(d_j) - \left(\sum_{u \in \mathcal{T}} a_u(d_i) \frac{\partial a_u(d_i)}{\partial w_{rs}} \right) \left(\sum_{t \in \mathcal{T}} \hat{a}_t(d_j) \hat{a}_t(d_i) \right)}{\|a(d_i)\|^2} \\ &\stackrel{(3)}{=} \frac{\sum_{t \in \mathcal{T}} \frac{\partial a_t(d_i)}{\partial w_{rs}} \|a(d_i)\| \hat{a}_t(d_j) - \sum_{u \in \mathcal{T}} a_u(d_i) \frac{\partial a_u(d_i)}{\partial w_{rs}} \operatorname{sim}(d_i,\, d_j)}{\|a(d_i)\|^2} \end{aligned} \quad (10)$$

Substituting t for u in the second term of (10) gives

$$\begin{aligned} Q(d_i,\, d_j) &= \frac{\sum_{t \in \mathcal{T}} \frac{\partial a_t(d_i)}{\partial w_{rs}} \left(\|a(d_i)\| \hat{a}_t(d_j) - a_t(d_i) \operatorname{sim}(d_i,\, d_j) \right)}{\|a(d_i)\|^2} \\ &= \frac{\sum_{t \in \mathcal{T}} \frac{\partial a_t(d_i)}{\partial w_{rs}} \left(\hat{a}_t(d_j) - \hat{a}_t(d_i) \operatorname{sim}(d_i,\, d_j) \right)}{\|a(d_i)\|} \end{aligned} \quad (11)$$

If $t \notin \mathit{Aff}(d_i) \setminus T(d_i)$ then $a_t(d_i)$ is by (1) constant and consequently $\frac{\partial a_t(d_i)}{\partial w_{rs}} = 0$. For $t \in \mathit{Aff}(d_i) \setminus T(d_i)$,

$$\begin{aligned} \frac{\partial a_t(d_i)}{\partial w_{rs}} &\stackrel{(1)}{=} \frac{1}{|Base_t(d_i)|} \sum_{t' \in Base_t(d_i)} \frac{\partial w_{t't} a_{t'}(d_i)}{\partial w_{rs}} \\ &= \frac{1}{|Base_t(d_i)|} \sum_{t' \in Base_t(d_i)} \begin{cases} a_r(d_i) & \text{if } (t',\, t) = (r,\, s) \\ w_{t't} \frac{\partial a_{t'}(d_i)}{\partial w_{rs}} & \text{otherwise.} \end{cases} \end{aligned} \quad (12)$$

The value of $\frac{\partial a_t(d_i)}{\partial w_{rs}}$ is computed recursively by (12). The recursion ends when $(t',\, t) = (r,\, s)$. Substituting from (12) into (11) and subsequently from (11) into (5) completes the formula.