Weighting Distributional Features for Automatic Semantic Classification of Words

Viktor Pekar* **Michael Krkoska†**
CompLing Group **Mentasys GmbH**
Univ. of Wolverhampton **Schönfeldstrasse 8**
Wolverhampton **76131 Karlsruhe**
WV1 1SB, UK **Germany**
v.pekar@wlv.ac.uk michael@mentasys.de

Abstract

The paper is concerned with weighting distributional features of words with the aim of improving their automatic semantic classification, a task relevant to a number of NLP applications such as lexicon acquisition or named entity recognition. The purpose of the paper is to bring attention to differences between two major weighting strategies: Discriminative Feature Weighting and Characteristic Feature Weighting. The comparative study includes three popular discriminative weighting functions (Mutual Information, Information Gain, and Gain Ratio), and three characteristic weighting functions (Term Strength, and the two newly introduced Local Term Strength and Confidence). We find that the two strategies, on the one hand, are characterized by their own optimal settings, and, on the other hand, similarly interact with the parameter optimization of the learning algorithm.

1 Introduction

Today many NLP applications employ the distributional approach to represent meanings of words. The approach seems especially useful in tasks where large lexical coverage is in question, since it represents meanings of words based only on cooccurrence statistics and thus eliminates the need in external lexical knowledge, such as a thesaurus or semantically annotated or sense-tagged corpus.

This approach represents the meaning of a word as a feature vector where each feature corresponds to a context of the word's use (e.g., another word or phrase that is syntactically related to it) found in a corpus. Using these arbitrary features, however, results in the fact that many of them are ambiguous or irrelevant, and significantly reduce the quality of the representation.

In the present paper, we study feature weighting methods, which aim to emphasize relevant features and downplay irrelevant ones, in application to distributional classification of words.

2 Two Feature Weighting Strategies

Let us assume that each word $n \in N$ of the training set is represented as a feature vector, consisting of features $f \in F$, each having value v_n^f, and that each n is assigned a class label $c \in C$, i.e. $\forall n \exists! c \in C : n \in c$. The general form of the feature weighting procedure can then be described as follows. For each f, the weighting function computes a weight $w(f, c)$, local to each training class $c \in C$. From such local weights of a feature, one may compute its single global weight, using some globalization technique. For example, as a global weight one can use the maximum local weight of f across all classes $w_{glob}(f) = max_{c \in C} w(f, c)$. After the weights have been applied to the training set, a classifier is learned and evaluated on the test set.

Text categorization literature distinguishes two major strategies that a function computing $w(f, c)$ can take. The first one embodies the assumption that greater score should be given to those features that better discriminate a particular class, i.e. tend to appear in instances of this class and not in instances of other classes (Discriminative Feature Weighting, DFW). A DFW function determines $w(f, c)$ from the distribution of f between c and $\bar{c}$, weighting highest those f that correlate with c most. Most scoring methods popular in text categorization (Chi-Square, Mutual Information, Information Gain, and others) are based on this strategy.

* The author was supported by RFBR grant#03-06-80008.

† The work was carried out as part of MSc thesis at the University of Karlsruhe.

The other strategy, by far less studied, assumes that most relevant features are those that are common to a set of similar instances and thus are most characteristic of them (Characteristic Feature Weighting, CFW). This strategy is exemplified in a function called Term Strength. As sets of similar instances needed to compute feature weights, it uses their clusters and thus is not supervised. In the present paper we propose two supervised functions of this type. As sets of similar instances, they use classes of words and determine $w(f, c)$ from the distribution of f across words $n \in c$, weighting greatest those f that are present with most n.

Functions that are based on the two different strategies are particularly interesting as an object of a comparative study, since they may display quite different behaviour, given the radically different ideas underlying them. In the present study we include Mutual Information, Information Gain and Gain Ratio[1], which instantiate the DFW strategy, and Term Strength and two newly proposed functions, which instantiate the CFW strategy. The particular problem we examine here is the choice between local and global variants of these functions.

The next section discusses these functions in more detail.

3 Feature Weighting Functions

3.1 Characteristic Feature Weighting

Term Strength (TS) was introduced in (Wilbur & Sirotkin 92) for improving efficiency of text categorization by feature selection. This method is based on the idea that most valuable features are shared by similar documents. It defines the weight of a feature as the probability of finding it in some document d given that it has also appeared in the document d', most similar to d. To calculate TS for feature f, for each word n we first determined its most similar word n' using a distributional similarity measure, thus preparing a set of pairs (n, n'). Pairing of words is asymmetric, i.e. one word can be the closest neighbor for several words. The TS weight for f was then calculated as the conditional probability of f appearing in n given that f appears also in n' (the ordering of words inside a pair being ignored):

$$TS(f) = p(f \in n \mid f \in n') \tag{1}$$

TS does not make use of the information about the words' distribution across classes and therefore is always global.

We introduce a supervised variant of TS (**TS-local**), which is different from TS in that, firstly, the most similar word for n is looked for not in the entire training set, but within the class of n; secondly, the weight for a feature is determined from the distribution of the feature across pairs of members of only that class:

$$TS_{loc}(f) = p(f \in n \mid f \in n') \ , (n, n') \in c \tag{2}$$

Thus, by weighting features using TS-local we aim to increase similarity between members of a class and disregard possible similarities across classes.

We further introduce another supervised CFW function, which we call **Confidence** of a feature (C) on the analogy with the notion from association rules mining (Agrawal *et al.* 93). The computed weight for feature f in class c may be described as the confidence of a rule, where the premise contains singular item set f and the consequence is c. We define Confidence of f as the proportion of instances in c which possess f to the total number of instances in c:

$$C(f, c) = \frac{\mid \{n \in c \mid f \in n\} \mid}{\mid \{n \in c\} \mid} \tag{3}$$

Unlike TS and TS-local, Confidence does not make use of the distributional similarity between words and is based solely on the semantic similarity encoded in class labels. As a set of similar words, it does not use pairs of words, but the entire word class, thereby capturing a wider range of similarities between its members. Note, however, that this can also be viewed as a disadvantage, since the method relies on the class to be homogeneous, i.e. it assumes there are no disjunct clusters of members inside the class, in which case indicative features would be largely obscured. This potential problem is counteracted by TS-local, which considers pairs of words inside a class rather than the class as a whole.

[1] We have experimented also with Chi-Square, Likelihood Ratio and Odds Ratio, finding that Mutual Information, Information Gain and Gain Ratio produce better results on our data. For a detailed report of these experiments, cf. (Krkoska 03).

3.2 Discriminative Feature Weighting

Mutual Information (MI) is an information-theoretic measure of association between two words, two tags or any other linguistic items, widely used in many statistical NLP applications. By computing pointwise MI between class c and feature f we measure how much the class label depends on the presence of the feature. Greater values of MI indicate that the feature is more predictive of class membership and thus should be preferred over features with smaller MI.

$$MI(f,c) = \log \frac{p(f,c)}{p(f)p(c)} \quad (4)$$

MI is known to give too high estimates for bigrams that involve rare words. We therefore experimented with a number of variants of MI that attempt to penalize less frequent words, but did not find that they improve on the traditional MI, which we explain by the nature of preprocessing of our data (see Section 4).

Information Gain (IG), is another well known feature weighting method, introduced into NLP from information theory. IG measures the relevance of feature f to the semantics of class c by computing the difference between the entropies of the class with and without the feature. It is different from MI in that it uses the fact of absence of a feature in a class and attributes greater weights to both features that tend to appear with its members and those that tend not to. IG[2] is defined as:

$$IG(f,c) = \sum_{d\in\{c,\overline{c}\}} \sum_{g\in\{f,\overline{f}\}} p(g,d) \log \frac{p(g,d)}{p(g)p(d)} \quad (5)$$

Gain Ratio (GR) is a normalized version of IG. GR aims to overcome one disadvantage of IG consisting in the fact that IG grows not only with the increase of dependence between f and c, but also with the increase of the entropy of f. That is why IG tends to assign smaller weights to those features that have low entropy in the training set, even though they are strongly correlated with certain classes. GR removes this factor by normalizing IG by the entropy of the feature:

$$GR(f,c) = \frac{IG(f,c)}{-\sum_{g\in\{f,\overline{f}\}} p(g)\log p(g)} \quad (6)$$

[2]Strictly speaking, the definition below does not define IG, but conditional entropy $H(c \mid f)$; the other ingredient of the IG function, the entropy of c, being constant and thus omitted from actual weight calculation.

The following sections present experimental results of evaluation of the six weighting functions.

4 Experimental Data

The evaluation was carried out on the task of classifying nouns into predefined classes. The meaning of each noun $n \in N$ was represented as a vector where features are verbs $v \in V$ with which the nouns are used as direct or prepositional objects. The values of the features were conditional probabilities $p(v|n)$. We used two datasets in our experiments: verb-noun co-occurrence pairs extracted from the British National Corpus (BNC) and from the Associated Press 1988 corpus (AP)[3]. Rare nouns were filtered out: the BNC data contained nouns that appeared with at least 5 different verbs and the AP data contained 1000 most frequent nouns, each of which appeared with at least 19 verbs.

To provide the extracted nouns with class labels needed for training and evaluation, the nouns were arranged into classes using WordNet in the following manner. Each class was made up of those nouns whose most frequent senses are hyponyms to a node seven edges below the root level of WordNet. Only those classes were used in the study that had 5 or more members. Thus, from the BNC data we formed 60 classes with 514 nouns and from the AP data 42 classes with 375 nouns. Assuming that relevant distributional data obtained from the corpora are reliable evidence about meanings of words as they are encoded in WordNet, we expect that application of more effective weighting methods would result in more accurate classification of test words into the created word classes.

During the experiments, classification was carried out by means of the k nearest neighbor classifier. This method has been shown to be quite robust on highly dimensional representations (e.g., (Yang & Pedersen 97)). In the present study we used the weighted k-nn algorithm (the vote of each neighbor was weighted by the score of its similarity to the test instance). To measure similarity between the vectors of nouns n and m we used the L1 distance:

$$L_1(n,m) = \sum_{v\in V} \mid p(v \mid n) - p(v \mid m) \mid \quad (7)$$

[3]Available from: http://www.cs.cornell.edu/home/llee/data/sim.html

We experimented with k = 1, 3, 5, 7, 10, 15, 20, 30, 50, 70, and 100. In the following sections, the results tables indicate the highest precision obtained among all k for a particular weighting method.

To evaluate the quality of classifications resulting from a particular weighting method, we used ten-fold cross-validation. The reported evaluation measure is precision obtained by microaveraging over the ten test sets.

As a baseline, we used the k-nn classifier trained on non-weighted instances.

5 Results

In carrying out either local or global weighting, one has the choice of weighting by the computed weights only training instances (e.g., (Mladenic 98)) or also test instances just before their classification (e.g., (Shankar & Karypis 00)). Table 1 presents the results of evaluation of the functions along two dimensions: (1) local versus global weighting and (2) weighted versus non-weighted test instances.

The results are similar on the two datasets. First, we see that global variants of the DFW functions outperform their local variants. This finding is consistent with that of (Debole & Sebastiani 03), who on the text categorization task found that global IG, GR and Chi-square perform better than their corresponding local schemas. We further see that in using MI and GR global weights, it is preferable to weight also test instances. The CFW functions, in contrast, demonstrate better effectiveness in their local variants (with the exception of TS-local on the BNC data). These functions also fare better when test instances are not weighted (including the global traditional TS).

The explanation for the DFW functions being better at global weighting while the CFW functions at local ones we see in the fact that with DFW, highest weighted are rare indicative features, which are likely to appear only in one class so that using the same weight for all classes does not cause confusion between them. With CFW, however, highest weighted are rather frequent features which are likely to be present in some other classes as well. Thus, when global weighting is carried out, all more or less frequent features of each class have approximately the same weight.

From Table 1 we also see that for all the five supervised functions, weighting test instances locally results in much poorer performance. After examining locally weighted test instances, we found that their representations usually contained an extremely large number of zeros. The reason for this is that when a test instance has many features uncommon with the particular training class with which it is being compared, these features receive zero weights, which eventually renders representations very sparse. This problem is eliminated when (1) one does not weight test instances at all or (2) when one weights them globally which guarantees that all features in a test instance that were present in the training set have a non-zero weight. To cope with the problem of zero weighted features in the locally weighted test instances, we tried three ways to smooth them:

(1) To the local weight $w(f,c)$ of f, we add the smallest of all weights for class c so that the new value $\tilde{v}$ of f in c is defined as:

$$\tilde{v}_n^f = v_n^f \cdot (w(f,c) + \min_{g \in \{g \in F | p(g,c) > 0\}} w(g,c))$$

(2) To the weighted value of f, we added its non-weighted value:

$$\tilde{v}_n^f = v_n^f \cdot w(f,c) + v_n^f$$

(3) In the case when f receives a zero local weight, it was replaced by its global weight.

Table 2 illustrates the results of the comparison of these smoothing methods.

As one can see, none of these smoothing methods improved the best performance of the functions. Although replacing zero local weight of a feature by its global weight resulted in the best local schemas for MI and GR (only on the BNC data), these schemas did not reach the accuracy of their purely global counterparts.

Figure 1 compares the performance of the best schema of each of the six weighting methods in relation to the number of nearest neighbors, for the BNC dataset.

From Figure 1, one can see that application of the functions (except TS) influences optimization of k. Compared to the baseline, the best k often shifts to lower values (e.g., from 20 for the baseline to 10 for GR, 7 for C, or 3 for TS-local).

	wgt	BNC						AP					
	test	MI	IG	GR	TS	TS_{loc}	C	MI	IG	GR	TS	TS_{loc}	C
loc	no	.3720	.3640	.3857	-	.3618	**.3973**	.3943	.3541	.3541	-	**.3992**	**.4285**
loc	yes	.2882	.2202	.3059	-	.1460	.1965	.2721	.1492	.2456	-	.2850	.3568
gl	no	.3797	**.3699**	.3857	**.3662**	**.3738**	.3699	.3921	**.3706**	**.3706**	**.3896**	.3895	.4024
gl	yes	**.4187**	.3565	**.4403**	.3114	.3138	.3485	**.4074**	.3626	**.3706**	.3866	.3690	.4052

Table 1: Local vs global variants of the functions. The baseline for BNC is .3796, for AP .3972.

Smoothing Method	BNC					AP				
	MI	IG	GR	TS_{loc}	C	MI	IG	GR	TS_{loc}	C
Add min.weight	.3740	.3582	.3857	.2310	.2472	.2723	.1492	.2456	.3252	.3702
Add non-weighted value	.3331	.3563	.3877	.3427	.3485	.3492	.3539	.3621	.3974	.3920
Replace by global	**.4074**	.3690	**.4131**	.3039	.3055	.3810	.3503	.3359	.3156	.4051

Table 2: Methods of smoothing locally weighted test instances.

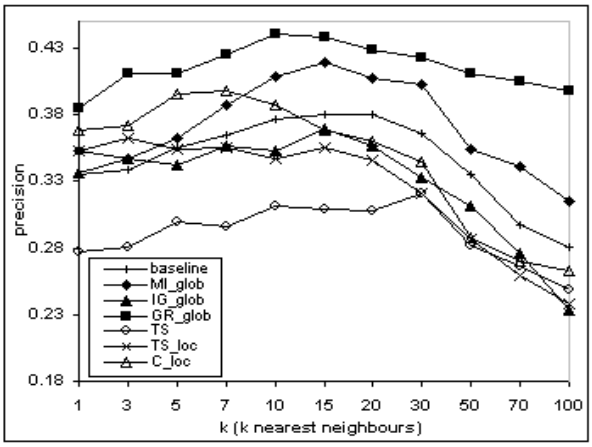

Figure 1: Performance of the best schemas of the weighting methods as functions of k on the BNC dataset.

6 Conclusion

In this paper we comparatively studied a number of feature weighting methods in application to the task of distributional classification of words. The study included functions which embody two distinct strategies of feature weighting - Discriminative Feature Weighting and Characteristic Feature Weighting. Our conclusions about relative merits of the functions can be summarized as follows.

On two different datasets, we found that weighting distributional features by means of MI, GR and Confidence, a newly proposed CFW function, almost always results in improvement of classification accuracy over the baseline. From the good performance of Confidence we conclude that characteristic features of a class can also serve as good class separators.

We found that the DFW functions perform significantly better in their global variants, while CFW functions are more effective in local ones. Moreover, when one carries out global weighting using MI or GR, it is advisable to also weight the test instances. In carrying out local weighting by means of TS-local and Confidence, effectiveness increases if one does not weight test instances.

Finally, we found that application of the weighting functions often interacts with optimization of k in k-nn, making the most optimal k shift to lower values.

References

(Agrawal *et al.* 93) R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. *ACM SIGMOD International Conference on Knowledge Discovery and Data Mining*, pages 207–216, 1993.

(Debole & Sebastiani 03) F. Debole and S. Sebastiani. Supervised term weighting for automated text categorization. *Proceedings of the 18th ACM Symposium on Applied Computing*, pages 784–788, 2003.

(Krkoska 03) M. Krkoska. Feature weighting for ontology extraction (in German). Unpublished M.Sc. thesis, AIFB, University of Karlsruhe, 2003.

(Mladenic 98) D. Mladenic. Feature subset selection in text learning. *Proceedings of the 10th European Conference on Machine Learning*, pages 95–100, 1998.

(Shankar & Karypis 00) S. Shankar and G. Karypis. A feature weight adjustment algorithm for document classification. *Proceedings of SIGKDD'00 Workshop on Text Mining*, 2000.

(Wilbur & Sirotkin 92) J.W. Wilbur and K. Sirotkin. The automatic identification of stopwords. *Journal of Information Science*, (18):45–55, 1992.

(Yang & Pedersen 97) Y. Yang and J.O. Pedersen. A comparative study on feature selection in text categorization. *Proceedings of the 14th International Conference on Machine Learning*, pages 412–420, 1997.