A Comparative Study of Text Preprocessing Approaches for Topic Detection of User Utterances

Roman Sergienko, Muhammad Shan and Wolfgang Minker
Institute of Telecommunication Engeneering, Ulm University
Albert-Einstein-Allee 43, 89081, Ulm, Germany
{roman.sergienko, muhammad.shan, wolfgang.minker}@uni-ulm.de

Abstract

The paper describes a comparative study of existing and novel text preprocessing and classification techniques for domain detection of user utterances. Two corpora are considered. The first one contains customer calls to a call centre for further call routing; the second one contains answers of call centre employees with different kinds of customer orientation behaviour. Seven different unsupervised and supervised term weighting methods were applied. The collective use of term weighting methods is proposed for classification effectiveness improvement. Four different dimensionality reduction methods were applied: stop-words filtering with stemming, feature selection based on term weights, feature transformation based on term clustering, and a novel feature transformation method based on terms belonging to classes. As classification algorithms we used *k*-NN and a SVM-based algorithm. The numerical experiments have shown that the simultaneous use of the novel proposed approaches (collectives of term weighting methods and the novel feature transformation method) allows reaching the high classification results with very small number of features.

Keywords: text classification, spoken language, term weighting

1. Introduction

An example of a spoken language processing application is spoken dialogue systems (SDS). A standard SDS consists of subsequent processing steps including (Minker and Bennacef, 2004): speech recognition, semantic text analysis, dialogue management, text generation, and speech synthesis.

The specific semantic model is designed for the problem domain of the SDS. Nowadays, multi-domain spoken dialogue systems constitute a breakthrough in the area of human-machine interfaces (Lee et al., 2009; Komatani et al., 2009). Multi-domain SDS include different specific semantic models for each problem domain. Prior to a detailed semantic analysis the general topic domain of user utterance needs to be determined. Domain detection can be formulated as a text classification problem based on a "bag-of-words" model. Designing domain-related semantic models after a domain detection step based on text classification may be more effective than designing a complex semantic model that applies for all domains.

Before text classification, it is necessary to perform text preprocessing which comprises three stages. The first one is the textual feature extraction based on raw document preprocessing. This includes procedures such as stop-words filtering (Fox, 1989) and stemming (Porter, 2001).

The second stage is the numerical feature extraction based on term weighting. The most well-known unsupervised term weighting method is TF-IDF (Salton and Buckley, 1988). The following supervised term weighting methods are also considered in our paper: Gain Ratio (GR) (Debole and Sebastiani, 2004), Confident Weights (CW) (Soucy and Mineau, 2005), Term Second Moment (TM2) (Xu and Li, 2007), Relevance Frequency (RF) (Lan et al., 2009), Term Relevance Ratio (TRR) (Ko, 2012), and Novel Term Weighting (NTW) (Gasanova et al., 2014a); these methods involve information about the classes of the documents. In this paper we propose collectives of term weighting methods that could improve classification effectiveness.

As a rule, the dimensionality for text classification problems is high even after stop-words filtering and stemming. Due to the high dimensionality, the classification may be inappropriately time-consuming, especially for real-time spoken dialogue systems. The third stage of text preprocessing is dimensionality reduction based on numerical features performing using feature selection or feature transformation. In addition to existing approaches, we propose a novel feature transformation method for text classification that significantly reduces dimensionality; the number of features will be equal to number of classes.

The classification algorithms we use the are *k*-NN algorithm and the SVM-based Fast Large Margin (Fan et al., 2008). A lot of investigations (Han et al., 2001; Baharudin et al., 2010; Joachims, 2002; Morariu et al., 2005) have shown effectiveness of the *k*-NN algorithm and SVM-based algorithms for text classification.

The main goal of our work is to perform a comparative study of text preprocessing techniques (term weighting and dimensionality reduction methods) and to investigate novel approaches (collectives of term weighting methods and the novel feature transformation method) for user utterance classification.

This paper is organized as follows: In Section 2, we describe the considered corpora. Section 3 describes the considered term weighting approaches. The dimensionality reduction methods for text classification are presented in Section 4. Section 5 describes the classification stage. The results of numerical experiments are presented in Section 6. Finally, we provide concluding remarks in Section 7.

2. Considered Corpora

We consider two different databases of user utterances in textual format after speech recognition.

Problem 1. The first corpus consists of 292,156 user utterances recorded in English from caller interactions with commercial automated agents. Utterances contain only one phrase for further routing. The database is provided by the company *Speech Cycle* (New York, USA). Utterances from this database are manually labelled by experts and divided into 20 classes (such as appointments, operator, bill, internet, phone and technical support). One of them is a special class TE-NOMATCH which includes utterances that cannot be put into another class or can be put into more than one class.

The database contains also 23,606 empty calls without any words. These calls were placed in the class TE-NOMATCH automatically and they were removed from the database. The average length of an utterance is 4.66 words, the maximal length is 19 words. There are a lot of identical utterances in the database; the corpus contains only 24,458 unique non-empty classified calls. The corpus is unbalanced.

For statistical analysis we performed 20 different divisions of the database into training and test samples randomly. This procedure was performed for two problem definitions separately. The train samples contain 90% of the calls and the test samples contain 10% of the calls. For each training sample we have designed a dictionary of unique words which appear in the training sample. The average size of the dictionary equals 3,304.

Problem 2. The second corpus contains 337 operator answers that were collected from a call service (Rafaeli et al., 2008). The answers were categorized into five classes of customer orientation behaviours: anticipating customer requests, offering explanations/justifications, educating the customer, providing emotional support, and offering personalized information. Such a user utterance classification problem may be also important in the field of spoken dialogue system design. Due to the small size of the second corpus, we have performed the Leave-One-Out (LOO) cross-validation for feature extraction, dimensionality reduction, and classification. The average dictionary size equals 802.

Therefore, we have one corpus with excess data and another corpus with scarce data for machine learning. It allows to test text classification approaches in different conditions.

3. Term Weighting Methods

As a rule, term weighting is a multiplication of two parts: the part based on the term frequency in a document (TF) and the part based on the term frequency in the whole training database. The TF-part is fixed for all considered term weighting methods and is calculated as following:

$$TF_{ij} = \log\left(tf_{ij} + 1\right); tf_{ij} = \frac{n_{ij}}{N_j},$$

where n_{ij} is the number of times the i^{th} word occurs in the j^{th} document, N_j is the document size (number of words in the document).

The second part of the term weighting is calculated once for each word from the dictionary and does not depend on an utterance for classification. We consider seven different methods for the calculation of the second part of term weighting.

3.1. Inverse Document Frequency (IDF)

IDF is a well-known unsupervised term weighting method which was proposed in (Salton and Buckley, 1988). There are some modifications of IDF and we use the most popular one:

$$idf_i = \log\frac{|D|}{n_i},$$

where $|D|$ is the number of documents in the training set and n_i is the number of documents that have the i^{th} word.

3.2. Gain Ratio (GR)

Gain Ratio (GR) is mainly used in term selection (Yang and Pedersen, 1997), but in (Debole and Sebastiani, 2004) it was shown that it could also be used for weighting terms. The definition of GR is as follows:

$$GR\left(t_i, c_j\right) = \frac{\sum_{c\in\{c_j,\bar{c}_j\}}\sum_{t\in\{t_j,\bar{t}_j\}} M\left(t,c\right)}{-\sum_{c\in\{c_j,\bar{c}_j\}} P\left(c\right)\cdot\log P\left(c\right)},$$

$$M\left(t,c\right) = P\left(t,c\right)\cdot\log\frac{P\left(t,c\right)}{P\left(t\right)\cdot P\left(c\right)},$$

where $P(t,c)$ is the relative frequency that a document contains the term t and belongs to the category c; $P(t)$ is the relative frequency that a document contains the term t and $P(c)$ is the relative frequency that a document belongs to category c. Then, the weight of the term t_i is the max value between all categories as follows:

$$GR\left(t_i\right) = \max_{c_j\in C} GR\left(t_i, c_j\right),$$

where C is a set of all classes.

3.3. Confident Weights (CW)

This supervised term weighting approach has been proposed in (Soucy and Mineau, 2005). Firstly, the proportion of documents containing term t is defined as the Wilson proportion estimate $p(x,n)$ by the following equation:

$$p\left(x,n\right) = \frac{x + 0.5z^2_{\alpha/2}}{n + z^2_{\alpha/2}},$$

where x is the number of documents containing the term t in the given corpus, n is the number of documents in the corpus and $\Phi\left(z_{\alpha/2}\right) = \alpha/2$, where Φ is the t-distribution (Student's law) when $n < 30$ and the normal distribution when $n \geq 30$.

In this work $\alpha = 0.95$ and $0.5z^2_{\alpha/2} = 1.96$ (as recommended by the authors of the method). For each term t and each class c two functions $p_{pos}(x,n)$ and $p_{neg}(x,n)$ are calculated. For $p_{pos}(x,n)$ x is the number of documents which belong to the class c and have term t; n is the number of documents which belong to the class c. For $p_{neg}(x,n)$ x is the number of documents which have the term t but do not belong to the class c; n is the number of documents which do not belong to the class c.

The confidence interval (p^-, p^+) at 0.95 is calculated using the following equation:

$$M = 0,5z^2_{\alpha/2}\sqrt{\frac{p(1-p)}{n+z^2_{\alpha/2}}}; p^- = p - M; p^+ = p + M.$$

The strength of the term t in the category c is defined as the follows:

$$str(t,c) = \begin{cases} \log_2 \frac{2p^-_{pos}}{p^-_{pos} + p^+_{neg}}, & \text{if } p^-_{pos} > p^+_{neg}, \\ 0, & \text{otherwise.} \end{cases}$$

The maximum strength (Maxstr) of the term t_i is calculated as follows:

$$Maxstr(t_i) = \max_{c_j \in C} str\left(t_i, c_j\right)^2.$$

3.4. Term Second Moment (TM2)

This supervised term weighting method was proposed in (Xu and Li, 2007). Let $P(c_j|t)$ be the empirical estimation of the probability that a document belongs to the category c_j with the condition that the document contains the term t; $P(c_j)$ is the empirical estimation of the probability that a document belongs to the category c_j without any conditions. The idea is the following: the more $P(c_j|t)$ is different from $P(c_j)$, the more important the term t_i is. Therefore, we can calculate the term weight as the following:

$$TM2(t_i) = \sum_{j=1}^{|C|} \left(P(c_j|t) - P(c_j)\right)^2,$$

where C is a set of all classes.

3.5. Relevance Frequency (RF)

The RF term weighting method was proposed in (Lan et al., 2009) and is calculated as the following:

$$rf(t_i, c_j) = \log_2\left(2 + \frac{a_j}{\max\{1, \bar{a_j}\}}\right),$$

$$rf(t_i) = \max_{c_j \in C} rf\left(t_i, c_j\right),$$

where a_j is the number of documents of the category c_j which contain the term t_i and $\bar{a_j}$ is the number of documents of all the other categories which also contain this term.

3.6. Term Relevance Ratio (TRR)

The TRR method (Ko, 2012) uses *tf* weights and it is calculated as the following:

$$TRR(t_i, c_j) = \log_2\left(2 + \frac{P(t_i|c_j)}{P(t_i|\bar{c_j})}\right),$$

$$P(t_i|c) = \frac{\sum_{k=1}^{|T_c|} tf_{ik}}{\sum_{l=1}^{|V|}\sum_{k=1}^{|T_c|} tf_{lk}},$$

$$TRR(t_i) = \max_{c_j \in C} TRR\left(t_i, c_j\right),$$

where c_j is a class of the document, $\bar{c_j}$ is all of the other classes of c_j, V is the vocabulary of the training data and T_c is the document set of the class c.

3.7. Novel Term Weighting (NTW)

This method was proposed in (Gasanova et al., 2014a). The details of the procedure are the following. Let L be the number of classes; n_i is the number of documents which belong to the i_{th} class; N_{ij} is the number of occurrences of the j_{th} word in all documents from the i_{th} class. $T_{ij} = N_{ij}/n_i$ is the relative frequency of occurrences of the j_{th} word in the i_{th} class; $R_j = \max_i T_{ij}$; $S_j = \arg\max_i T_{ij}$ is the class which we assign to the j_{th} word. The term relevance C_j is calculated by the following:

$$C_j = \frac{1}{\sum_{i=1}^{L} T_{ij}} \cdot \left(R_j - \frac{1}{L-1} \cdot \sum_{i=1, i \neq S_j}^{L} T_{ij}\right).$$

3.8. Collectives of Term Weighting Methods

As a novel approach we propose collectives of term weighting methods. In this case we organize a meta-classifier based on majority vote. The majority vote procedure does not require additional learning. We have designed the collectives with different numbers of included methods from 7 to 3, with consistent exception of the worst methods.

4. Dimensionality reduction methods

4.1. Stop-word filtering and stemming

We consider stop-word filtering with stemming as a language-based dimensionality reduction method. It is performed before numerical feature extraction. We used special libraries ("tm", "SnowballC") in the programming language *R* for stop-word filtering and stemming for English.

4.2. Feature selection based on term weights

Term weighting methods provide a natural feature selection as it is possible to ignore terms with the lowest weights.
For RF, TM2, and TRR methods we decreased the dictionary size from 100% to 10% with the interval equals 10. This means deleting the corresponding number of the terms with the lowest weights. IDF and NTW provide getting a lot of terms with the equal highest value. For IDF the highest weight means that the term occurs only in one document from the training sample, for NTW it means that the term occurs only in documents of one class. Therefore, for these two methods we used different constraints for the value of weights. CW and GR provide getting a lot of terms with zero weights; it means that these two methods provide feature selection automatically. For the first problem we obtain 43.5% of the dictionary as terms with non-zero weights for GR and 20.4% for CW on the average; for the second problem 43.6% and 6.0% correspondingly. We also decreased the size of the dictionary for CW and GR with the class-based approach only for problem 1.

4.3. Feature transformation based on term clustering

The idea of using class-based language model by applying term clustering was proposed in (Momtazi and Klakow, 2009). It is possible to use the term clustering in a dictionary for dimensionality reduction. In this case we suggest preprocessing our dictionary such that words of equal or

similar weights are placed in the same cluster and one common weight (a new feature) will be assigned to all words in this cluster.

In our study we use class-based term clustering by weights as described in (Gasanova et al., 2014b). Term clustering is performed for each class separately. Therefore, it is necessary to assign each term from the dictionary to one corresponding class. During supervised term weighting methods CW, GR, RF, NTW, and TRR such an assignment is performed automatically (see Section 3). With IDF and TM2 we can also assign one class for each term using the maximal relative frequency of the word in classes:

$$S_j = \arg\max_{c \in C} \frac{n_{jc}}{N_c},$$

where S_j is the most appropriate class for the j^{th} term, c is an index of a class, C is a set of all classes, n_{jc} is number of documents of the c^{th} class which contain the j^{th} term, N_c is the number of all documents of the c^{th} class.

In order to reduce the dictionary size we apply hierarchical agglomerative clustering (Ward Jr, 1963) with Euclidean metric. As a common weight of the cluster we calculate the arithmetic mean of all term weights from this cluster. We set the maximal number of clusters for each class 10, 20, 50 and 100.

The details of the feature transformation based on term clustering are the following:

1. Assign each term from the dictionary of the text classification problem to the most appropriate class.
2. For each class perform:
2.1. Set the maximal number of clusters *N*
2.2. Start with disjointed terms (each term a_i of the current class forms its own cluster c_i).
2.3. Calculate all distances between pairs of clusters. In our case distance d_{ij} between i^{th} and j^{th} clusters equals to: $d_{ij} = |T_i - T_j|$, where T_i and T_j are weights of corresponding clusters.
2.4. Find two closest clusters c_i and c_j ($i < j$).
2.5. Add cluster c_j to cluster c_i. Calculate the new weight of the joined cluster as arithmetical weight mean of all terms that belong to the new cluster. Increment *i*, $i = i + 1$.
2.6. Recalculate the distances between new cluster and other clusters.
2.7. If number of clusters less than *N* go to step 2.4. Otherwise END

4.4. Novel feature transformation based on terms belonging to classes

We propose a novel feature transformation method based on terms belonging to classes. After the assigning of each term to one class (see Section 4.3), we can calculate the sums of term weights in a document for each class separately. We can consider these sums as new features of the text classification problem. Therefore, such a method significantly reduces the dimensionality: it equals the number of classes. The details of the method are the following:

1. Assign each term from the dictionary of the text classification problem to the most appropriate class:
2. Give the document *D* for classification.
3. Put $S_i = 0$, *i*=1..*C*, where *C* is the number of classes (categories).
4. For each term *t* in the document *D* do:
4.1. $S_i = S_i + w_t$, where *i* is the class of the t_{th} term in correspondence with the assignment on the step 1, w_t is the weight of the t_{th} term.
5. Put S_i, *i*=1..*C* as transformed features of the text classification problem.

5. Classification algorithms

For classification we use the *k*-NN algorithm with weight distance (*k*) and the SVM-based algorithm Fast Large Margin (SVM-FLM) (Fan et al., 2008). *RapidMiner* with standard setting (Shafait et al., 2010) was used as software for classification algorithm application. The classification criterion is the macro *F*-score (Goutte and Gaussier, 2005) which is appropriate for classification problems with unbalanced classes. For *k*-NN we performed validation of *k* from 1 to 15 on the validation sample. We used 80% of the train sample for the first level of learning and 20% for the validation.

6. Results of numerical experiments

Tables 1-4 show the results of the numerical experiments for problems 1 and 2 with three situations: without dimensionality reduction (all terms are used), with stop-words filtering + stemming, and with the novel feature transformation method (novel FT). The procedure with stop-words filtering and stemming was not combined with other dimensionality reduction methods. For all situations the ranking of term weighting methods was performed with *t*-test (the confidence probability equals 0.95). The ranks are illustrated in brackets. Other comparisons were also performed with *t*-test. An asterisk * denotes the number of the best term weighting methods in the collectives. The best results in tables are bold.

Term weighting method	***F*-score**		
	All terms	**Stop-word+ stemming**	**Novel FT**
IDF	0.855 (6-8)	0.777 (7)	0.819 (7)
GR	0.851 (6-8)	0.766 (8)	0.841 (6)
CW	0.870 (2-4)	0.784 (4-6)	0.851 (3-4)
RF	0.855 (6-8)	0.783 (4-6)	0.849 (5)
TM2	0.865 (5)	0.784 (4-6)	0.853 (3-4)
TRR	0.873 (2-4)	**0.793 (1-2)**	0.862 (2)
NTW	0.871 (2-4)	0.789 (3)	0.844 (5)
Collective	**0.883 (1)*7**	**0.799 (1-2)*7**	**0.877 (1)*7**

Table 1: Results for problem 1 with *k*-NN

The results of feature selection are presented in Figures 1-2, the results for feature transformation based on term clustering are illustrated in Figures 3-4.

The results show the effectiveness of the proposed collectives of term weighting methods. The best classification results for both problems are obtained with collectives with all words (the collective with seven methods for problem 1 and with six methods for problem 2).

Term weighting method	*F*-score		
	All terms	Stop-word+ stemming	Novel FT
IDF	**0.873 (1)**	**0.836 (1)**	0.544 (8)
GR	0.670 (8)	0.680 (7)	0.621 (5-7)
CW	0.835 (5)	0.749 (6)	0.747 (3-4)
RF	0.864 (3-4)	0.819 (4)	0.744 (5-7)
TM2	0.734 (7)	0.720 (7)	0.618 (3-4)
TRR	0.865 (3-4)	0.823 (2-3)	**0.792 (1)**
NTW	0.825 (6)	0.797 (5)	0.621 (5-7)
Collective	0.867 (2)*3	0.823 (2-3)*4	0.773 (2)*3

Table 2: Results for problem 1 with SVM-FLM

Term weighting method	*F*-score		
	All terms	Stop-word+ stemming	Novel FT
IDF	0.481 (5)	0.461 (5)	0.490 (4)
GR	0.416 (8)	0.395 (8)	0.182 (8)
CW	0.466 (7)	0.399 (7)	0.345 (7)
RF	0.499 (2)	0.496 (3)	**0.547 (1)**
TM2	0.475 (6)	0.458 (6)	0.480 (5)
TRR	0.489 (4)	0.510 (1-2)	0.527 (3)
NTW	0.497 (3)	0.483 (4)	0.471 (6)
Collective	**0.540 (1)*6**	**0.521 (1)*5**	0.543 (2)*3

Table 3: Results for problem 2 with *k*-NN

Term weighting method	*F*-score		
	All terms	Stop-word+ stemming	Novel FT
IDF	0.584 (2)	0.531 (2-3)	0.517 (3)
GR	0.478 (5)	0.378 (7)	0.211 (8)
CW	0.346 (8)	0.300 (8)	0.224 (7)
RF	0.575 (3)	0.531 (2-3)	**0.555 (1)**
TM2	0.404 (7)	0.452 (6)	0.487 (5)
TRR	0.548 (4)	0.512 (4)	0.506 (4)
NTW	0.464 (6)	0.471 (5)	0.481 (6)
Collective	**0.588 (1)*6**	**0.541 (1)*6**	0.535 (2)*7

Table 4: Results for problem 2 with SVM-FLM

Stop-words filtering with stemming results in a significant decrease in classification effectiveness. A similar situation is observed when applying feature selection; we do not observe a statistically significant decrease of *F*-score only for GR (one of the worst methods) and for TRR with 90% of the dictionary. This means that useful information is lost. The reason lies in the fact that short utterances for classification are used.

The most effective dimensionality reduction method is feature transformation based on term clustering. We obtain the best result for problem 1 only with 1532 features (TRR, *k*-NN); for problem 2 it is possible to increase classification effectiveness with feature transformation based on term clustering.

The novel feature transformation method provides appro-

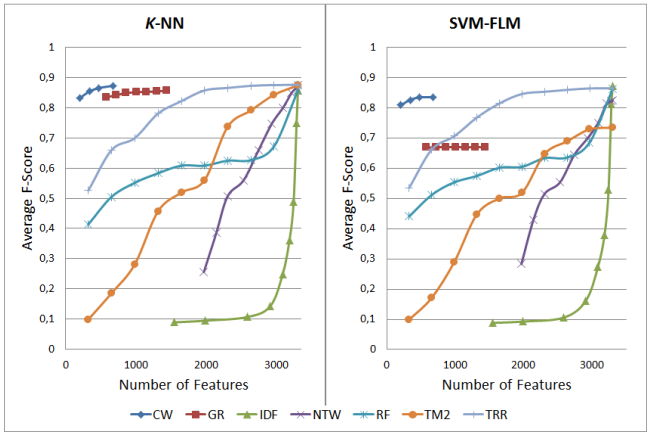

Figure 1: Feature selection for problem 1.

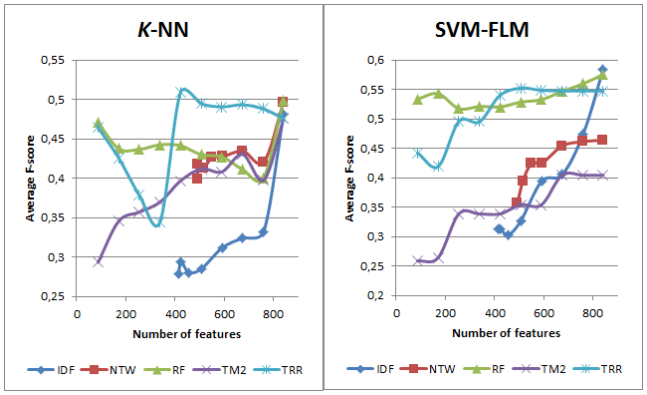

Figure 2: Feature selection for problem 2.

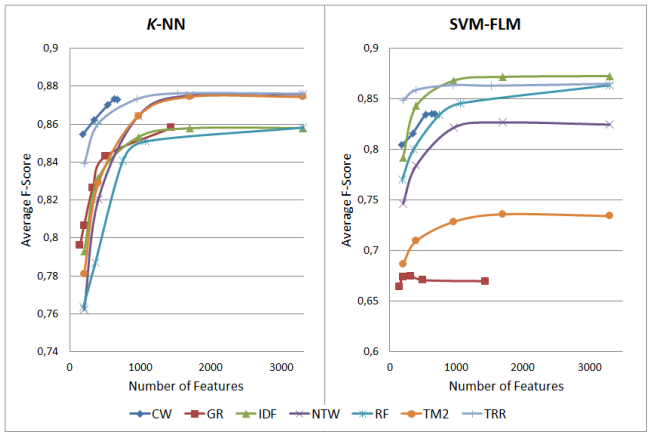

Figure 3: Feature transformation based on term clustering for problem 1.

priate classification results with a very compact feature set. For the first problem with *k*-NN the novel FT with collectives of term weighting methods provides the better result, than all other cases excluding the collective of term weighting methods with all words. Therefore, the novel FT is appropriate for real-time systems.

7. Conclusion

In the described work, we have shown that the collectives of term weighting methods may improve classification effec-

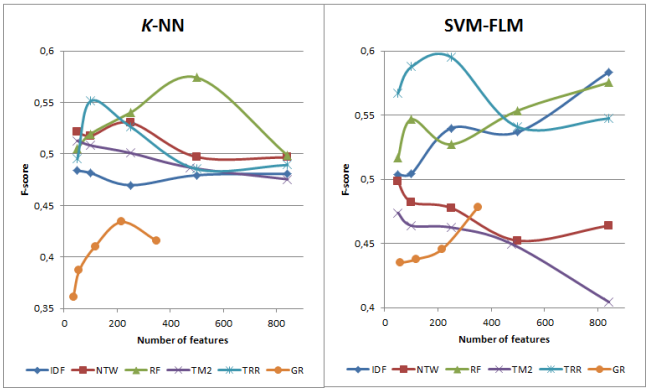

Figure 4: Feature transformation based on term clustering for problem 2.

tiveness for domain detection of user utterances. The best dimensionality reduction method is feature transformation based on term clustering which is able to decrease dimensionality significantly without decrease of the classification effectiveness. The novel feature transformation method reduces the dimensionality radically with appropriate classification results and can be useful for real-time classification systems.

8. Bibliographical References

Baharudin, B., Lee, L. H., and Khan, K. (2010). A review of machine learning algorithms for text-documents classification. *Journal of advances in information technology*, 1(1):4–20.

Debole, F. and Sebastiani, F. (2004). Supervised term weighting for automated text categorization. In *Text mining and its applications*, pages 81–97. Springer.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874.

Fox, C. (1989). A stop list for general text. In *ACM SIGIR Forum*, volume 24, pages 19–21. ACM.

Gasanova, T., Sergienko, R., Akhmedova, S., Semenkin, E., and Minker, W. (2014a). Opinion mining and topic categorization with novel term weighting. In *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, ACL 2014*, pages 84–89.

Gasanova, T., Sergienko, R., Semenkin, E., and Minker, W. (2014b). Dimension reduction with coevolutionary genetic algorithm for text classification. In *Informatics in Control, Automation and Robotics (ICINCO), 2014 11th International Conference on*, volume 1, pages 215–222. IEEE.

Goutte, C. and Gaussier, E. (2005). A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In *Advances in information retrieval*, pages 345–359. Springer.

Han, E.-H. S., Karypis, G., and Kumar, V. (2001). *Text Categorization Using Weight Adjusted k-Nearest Neighbor Classification*. Springer.

Joachims, T. (2002). *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms*. Kluwer Academic Publishers.

Ko, Y. (2012). A study of term weighting schemes using class information for text classification. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 1029–1030. ACM.

Komatani, K., Kanda, N., Nakano, M., Nakadai, K., Tsujino, H., Ogata, T., and Okuno, H. G. (2009). Multi-domain spoken dialogue system with extensibility and robustness against speech recognition errors. In *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue*, pages 9–17. Association for Computational Linguistics.

Lan, M., Tan, C. L., Su, J., and Lu, Y. (2009). Supervised and traditional term weighting methods for automatic text categorization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(4):721–735.

Lee, C., Jung, S., Kim, S., and Lee, G. G. (2009). Example-based dialog modeling for practical multi-domain dialog system. *Speech Communication*, 51(5):466–484.

Minker, W. and Bennacef, S. (2004). *Speech and human-Machine dialog*. Springer Science & Business Media.

Momtazi, S. and Klakow, D. (2009). A word clustering approach for language model-based sentence retrieval in question answering systems. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1911–1914. ACM.

Morariu, D. I., Vintan, L. N., and Tresp, V. (2005). Meta-classification using svm classifiers for text documents. *Intl. Jrnl. of Applied Mathematics and Computer Sciences*, 1(1).

Porter, M. F. (2001). Snowball: A language for stemming algorithms.

Rafaeli, A., Ziklik, L., and Doucet, L. (2008). The impact of call center employees' customer orientation behaviors on service quality. *Journal of Service Research*, 10(3):239–255.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523.

Shafait, F., Reif, M., Kofler, C., and Breuel, T. M. (2010). Pattern recognition engineering. In *RapidMiner Community Meeting and Conference*, volume 9. Citeseer.

Soucy, P. and Mineau, G. W. (2005). Beyond tfidf weighting for text categorization in the vector space model. In *IJCAI*, volume 5, pages 1130–1135.

Ward Jr, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244.

Xu, H. and Li, C. (2007). A novel term weighting scheme for automated text categorization. In *Intelligent Systems Design and Applications, 2007. ISDA 2007. Seventh International Conference on*, pages 759–764. IEEE.

Yang, Y. and Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. In *ICML*, volume 97, pages 412–420.