Verbal Intelligence Identification Based on Text Classification

Roman Sergienko and Alexander Schmitt

Institute of Communications Engineering, Ulm University, Ulm, Germany

{roman.sergienko, alexander.schmitt}@uni-ulm.de

Abstract

This paper analyses and compares term weighting methods for automatic verbal intelligence identification from speech. Two different corpora are used; the first one contains monologues on the same topic; the second one contains dialogues between two or three people. The problem is described as a text classification task with two classes: low and high verbal intelligence. Seven different term weighting methods were applied for text classification using the *k*-NN algorithm. The best result is obtained with the Confident Weights method as a term weighting method for the dialogue corpus. The best classification accuracy equals 0.80 and the best macro F1-score equals 0.79. The numerical results have shown that highest scores can be obtained when using a very small number of terms which characterize only the class of higher verbal intelligence.

Index Terms: verbal intelligence, text classification, term weighting, confident weights.

1. Introduction

One of the important speaker traits is verbal intelligence. Verbal intelligence (VI) is the ability to use language for accomplishing certain goals [1]. Automatic identification of verbal intelligence can make spoken dialogue systems (SDS) more helpful and user-friendly because such a SDS can adapt its content and complexity of a dialogue in accordance with the level of the verbal intelligence of the user.

One of the ways for automatic verbal intelligence recognition is to consider a dictionary that a person uses in speech. This approach is based on an idea that people with different levels of VI use different words or employ them with different frequency. Therefore, we formulate the verbal intelligence identification problem as a text classification task.

We consider two corpora for verbal intelligence identification which are presented in [2]. The first one contains monologues on the same topic; the second one contains dialogues between two or three people. The problem is described as a text classification task with two classes: low and high verbal intelligence.

In the vector space model [3] text classification is considered as a machine learning problem. The complexity of text categorization with a vector space model is compounded by the need to extract numerical data from text information before applying machine learning algorithms. Therefore, text classification consists of two parts: text preprocessing and classification algorithm application using the obtained numerical data.

All text preprocessing methods are based on the idea that the category of the document depends on the words or phrases from this document. One of the most popular models for document representation is the "bag-of-words" model, in which the word order is ignored. For numerical feature extraction term weighting methods are applied. The most well-known unsupervised term weighting method is TF-IDF [4]. The following supervised term weighting methods are also considered in the paper: Gain Ratio (GR) [5], Confident Weights (CW) [6], Term Second Moment (TM2) [7], Relevance Frequency (RF) [8], Term Relevance Ratio (TRR) [9], and Novel Term Weighting (NTW) [10]; these methods involve information about the classes of the documents.

As a classification algorithm we propose a *k*-NN algorithm. Some investigations [11, 12, 13] have shown the effectiveness of the *k*-NN algorithm for text classification.

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

2. Corpora description

The corpora for verbal intelligence identification are described in [2]. The first corpus consists of German native speakers monologues and the second one consists of dialogues collected at the University of Ulm, Germany. The monologues are descriptions of two short films; the dialogues are discussions about problems of German education. The first corpus contains 100 monologues of 100 different speakers. The second corpus contains 53 dialogues of 91 different speakers. 52 dialogues are conversations between two persons and one dialogue is a conversation between three persons. The set of the dialogue speakers is a subset of the set of the monologue speakers. The corpora contain audio files and also text documents after speech recognition. The data corpora contain the verbal intelligence quotients of each speaker, which were measured with the Hamburg Wechsler Intelligence Test for Adults [14]. The speakers are from different social groups with different education level.

The speakers were clustered into two classes with the *K*-means algorithm [2]. The first class means lower verbal intelligence and the second one means higher verbal intelligence. The corpus of the monologues contains 40 speakers with lower verbal intelligence and 60 speakers with higher verbal intelligence. The corpus of the dialogues contains 37 speakers with lower verbal intelligence and 54 speakers with higher verbal intelligence. For each speaker from the corpus of the dialogues, all his phrases were extracted from the dialogues and were placed into the same file because a speaker can be involved in more than one dialogue. Therefore, 53 dialogues were transformed to 91 documents, each of them corresponds to one speaker.

Due to the small size of the corpora we used Leave-One-Out (LOO) cross validation for feature extraction, feature selection, and classification.

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 7 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 [4]. 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 [15], but in [5] it was shown that it could also be used for weighting terms, since its value reflects the importance of a term. 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)

The Confident Weights method (CW) is a supervised term weighting that involves information about classes which correspond to documents. This approach has been proposed in [6].

Firstly, the proportion of documents containing term t is defined as the Wilson proportion estimate $p(x,n)$ [16] 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 (Students 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 equations:

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

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

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

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

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

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

The numerical experiments conducted in [6] have shown that the CW method outperforms the Gain Ratio and TF-IDF with SVM and *k*-NN as classification methods on three benchmark corpora.

3.4. Term Second Moment (TM2)

This supervised term weighting method was proposed in [7].

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 [8] and is calculated as the following:

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

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

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 [9] uses *tf* weights and it is calculated as the following:

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

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

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

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 [10, 17]. Term weight is calculated using a modified formula of fuzzy rules relevance estimation for fuzzy classifiers [18]. The membership function has been replaced by word frequency in the current class.

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:

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

The value of weights varies from 0 to 1. The maximal weight means that the term occurs only in one class, the minimal weight means that the term occurs in all classes with the same frequency.

4. Numerical experiments

For feature extraction, feature selection, and classification we used Leave-One-Out (LOO) cross validation with both corpora. The first stage of the numerical experiments is dictionary forming based on the training documents. The dictionary size varies from 3092 to 3183 for the monologue corpus and from 6892 to 7082 for the dialogue corpus. After that term weighting with 7 different methods was performed. As a classification algorithm we applied *k*-NN algorithm with weighted vote. We varied *k* from 1 to 30. *RapidMiner* was used as software for *k*-NN application [19]. As classification criteria we used macro F1-score and accuracy [20].

Tables 1 and 2 show the best classification results without feature selection (with all terms) for the monologue and dialogue corpora correspondingly.

Table 1: *Classification results without feature selection for monologues.*

Method	Macro F1-score		Accuracy	
	Value	Best *k*	Value	Best *k*
IDF	0.63	7	0.62	7
GR	0.60	18	0.63	18
CW	0.62	22	0.63	22
TM2	0.43	3	0.60	5
RF	0.58	7	0.62	7
TRR	0.59	19	0.63	19
NTW	0.59	6	0.63	6

Table 2: *Classification results without feature selection for dialogues.*

Method	Macro F1-score		Accuracy	
	Value	Best *k*	Value	Best *k*
IDF	0.64	7	0.62	7
GR	0.54	2	0.59	3
CW	**0.79**	7	**0.80**	7
TM2	0.37	1	0.59	1
RF	0.54	4	0.59	4
TRR	0.56	3	0.60	3
NTW	0.56	1	0.60	1

Term weighting methods provide a natural feature selection method; it is possible to ignore terms with the lowest weights. Therefore, we performed feature selection for different term weighting methods with predefined constraint for the minimal weight value. These constraints depend on the variety intervals for different term weighting methods. Tables 3-7 contain the best classification results for feature selection for IDF, TM2, RF, TRR, and NTW. For GR and CW feature selection was not performed because the GR and CW methods provide a very small number of terms with non-zero weights for the considered corpora automatically.

Table 3: *Feature selection for IDF.*

Constraint	-	1.0	2.0	3.0	4.0
Monologues, F1-score,	0.63	0.63	**0.67**	0.48	0.59
Monologues, accuracy,	0.62	0.61	0.65	0.55	0.43
Dialogues, F1-score,	0.64	0.64	0.64	0.46	0.63
Dialogues, accuracy,	0.62	0.62	0.62	0.60	0.61

The numerical experiments showed that feature selection can increase classification effectiveness. The best F-score for the monologue corpus equals 0.67 by the IDF and RF methods with feature selection and the best accuracy value for the monologue corpus equals 0.67 by the NTW method with feature selection. These values are relatively low for the two-classes classification problem. The best results for the dialogue corpus are better. The best F-score for the dialogue corpus equals 0.79 and the best classification accuracy equals 0.80 by the CW method. These values are encouraging.

Table 4: *Feature selection for TM2.*

Constraint	-	0.01	0.1	0.3
Monologues, F1-score,	0.43	0.43	0.43	0.43
Monologues, accuracy,	0.60	0.60	0.60	0.60
Dialogues, F1-score,	0.37	0.38	0.38	0.38
Dialogues, accuracy,	0.59	0.60	0.60	0.60

Table 5: *Feature selection for RF.*

Constraint	-	1.6	2.0
Monologues, F1-score,	0.58	0.59	**0.67**
Monologues, accuracy,	0.62	0.63	0.65
Dialogues, F1-score,	0.54	0.63	0.70
Dialogues, accuracy,	0.59	0.61	0.71

Table 6: *Feature selection for TRR.*

Constraint	-	1.0	1.5	2.0
Monologues, F1-score,	0.59	0.58	0.49	0.65
Monologues, accuracy,	0.63	0.62	0.60	0.66
Dialogues, F1-score,	0.56	0.53	0.67	0.64
Dialogues, accuracy,	0.60	0.60	0.67	0.63

Table 7: *Feature selection for NTW.*

Constraint	-	0.5	0.6	0.7	1.0
Monologues, F1-score,	0.59	0.65	0.57	0.60	0.59
Monologues, accuracy,	0.63	**0.67**	0.52	0.43	0.41
Dialogues, F1-score,	0.56	0.67	0.65	0.72	0.64
Dialogues, accuracy,	0.60	0.66	0.66	0.71	0.61

We can formulate two reasons why the results for the dialogue corpus are better than for the monologue corpus. At first, the documents for the dialogue corpus are larger and the dictionary is twice as large as for the monologue corpus. Therefore, we have more linguistic information for verbal intelligence identification. As a second explanation we can suppose that verbal intelligence is expressed clearer during a dialogue.

The best result for the dialogue corpus is obtained by the Confident Weights method (CW). This method provides getting zero values for a lot of terms automatically. In our case the number of terms with non-zero values are extremely low. For the dialogue corpus the average number of terms with non-zero weights equals 4.7 (min = 3, max = 6). The CW methods also determines the most appropriate class for each term from the dictionary automatically. For the dialogue corpus all terms with non-zero values belong to the second class which means higher verbal intelligence.

In the situation with extremely small number of terms with non-zero values, some documents have all features with zero values. The classification algorithm defines such a document as an element of the first class (lower verbal intelligence) automatically. Therefore, the best term weighting method (CW) for the dialogue corpus determines very small number of words that characterize only the class of higher verbal intelligence.

We can explain why we get a very small number of terms with non-zero values. These terms must satisfy two contradictory conditions:

1) The terms must be well-known and in general use. Specific words (i.e. professional terms) can be used by a restricted number of people even with high verbal intelligence. This condition is especially critical for our very small corpus with not more than 100 different speakers.

2) The terms must characterize high verbal intelligence.

In our case such words are ”missing”, ”higher”, ”syllabus” (translation from German).

We suppose that larger corpora for verbal intelligence identification than the considered ones can allow to increase the number of significant terms and classification effectiveness. It is possible to use written text for larger corpora creating. Maybe it would be also possible to increase the number of classes for categorising verbal intelligence.

5. Conclusions

In this paper we have presented the effectiveness of a comprehensive set of term weighting methods on the problem of verbal intelligence recognition. Two corpora for verbal intelligence identification were considered; the first one contains monologues on the same topic; the second one contains dialogues between two or three people. The problem is described as a text classification task with two classes: low and high verbal intelligence. Seven different term weighting methods were applied for text classification using the k-NN algorithm. Feature selection was also performed.

The numerical results showed relatively low classification effectiveness for the monologue corpus; the best F-score equals 0.67 and the best classification accuracy equals 0.67. The results for the dialogue corpus are better. The best classification accuracy equals 0.80 and the best F-score equals 0.79 with the Confident Weights method. The Confident Weights method provides getting extremely small number of terms with non-zero values which characterize only the class of higher verbal intelligence.

As a general conclusion we can formulate that automatic verbal intelligence identification based on text classification is a sophisticated problem, especially for small corpora. We suppose that the extension of the corpora for verbal intelligence identification can improve classification effectiveness.

6. References

[1] A. T. Cianciolo and R. J. Sternberg, *Intelligence: A brief history*. John Wiley & Sons, 2008.

[2] K. Zablotskaya, S. Walter, and W. Minker, "Speech data corpus for verbal intelligence estimation." in *LREC*, 2010.

[3] F. Sebastiani, "Machine learning in automated text categorization," *ACM computing surveys (CSUR)*, vol. 34, no. 1, pp. 1–47, 2002.

[4] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information processing & management*, vol. 24, no. 5, pp. 513–523, 1988.

[5] F. Debole and F. Sebastiani, "Supervised term weighting for automated text categorization," in *Text mining and its applications*. Springer, 2004, pp. 81–97.

[6] P. Soucy and G. W. Mineau, "Beyond tfidf weighting for text categorization in the vector space model," in *IJCAI*, vol. 5, 2005, pp. 1130–1135.

[7] H. Xu and C. Li, "A novel term weighting scheme for automated text categorization," in *Intelligent Systems Design and Applications, 2007. ISDA 2007. Seventh International Conference on*. IEEE, 2007, pp. 759–764.

[8] M. Lan, C. L. Tan, J. Su, and Y. Lu, "Supervised and traditional term weighting methods for automatic text categorization," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 4, pp. 721–735, 2009.

[9] Y. Ko, "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*. ACM, 2012, pp. 1029–1030.

[10] T. Gasanova, R. Sergienko, S. Akhmedova, E. Semenkin, and W. Minker, "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*, 2014, pp. 84–89.

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

[12] O.-W. Kwon and J.-H. Lee, "Text categorization based on k-nearest neighbor approach for web site classification," *Information Processing & Management*, vol. 39, no. 1, pp. 25–44, 2003.

[13] B. Baharudin, L. H. Lee, and K. Khan, "A review of machine learning algorithms for text-documents classification," *Journal of advances in information technology*, vol. 1, no. 1, pp. 4–20, 2010.

[14] D. Wechsler, *Handanweisung zum Hamburg-Wechsler-Intelligenztest für Erwachsene,(HAWIE)*. Huber, 1982.

[15] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *ICML*, vol. 97, 1997, pp. 412–420.

[16] E. B. Wilson, "Probable inference, the law of succession, and statistical inference," *Journal of the American Statistical Association*, vol. 22, no. 158, pp. 209–212, 1927.

[17] R. Sergienko, T. Gasanova, E. Semenkin, and W. Minker, "Text categorization methods application for natural language call routing," in *Informatics in Control, Automation and Robotics (ICINCO), 2014 11th International Conference on*, vol. 2. IEEE, 2014, pp. 827–831.

[18] H. Ishibuchi, T. Nakashima, and T. Murata, "Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 29, no. 5, pp. 601–618, 1999.

[19] F. Shafait, M. Reif, C. Kofler, and T. M. Breuel, "Pattern recognition engineering," in *RapidMiner Community Meeting and Conference*, vol. 9. Citeseer, 2010.

[20] C. Goutte and E. Gaussier, "A probabilistic interpretation of precision, recall and f-score, with implication for evaluation," in *Advances in information retrieval*. Springer, 2005, pp. 345–359.