Aicyber at SemEval-2016 Task 4: i-vector based sentence representation

Steven Du, Zhang Xi
www.aicyber.com
No. 39 Nanjing Road, Capitaland International Trade Center Block B 1604
Tianjin China
{steven,zhangxi}@aicyber.com

Abstract

This paper introduces aicyber's systems for SemEval 2016 , Task 4A. The first system is build on vector space model (VSM), the second system is build on a new framework to estimate sentence vector, it is inspired by the i-vector in speaker verification domain. Both systems are evaluated on SemEval 2016 (Task4A) as well as IMDB dataset. Evaluation results show that the i-vector based sentence vector is an alternative approach to present sentence.

1 Introduction

The SemEval 2016 Task 4 is sentiment analysis in tweets. The subtask, task A focused on classifying tweets into three classes: positive, negative or neutral sentiment (Nakov et al., 2016).

This paper will first presents the submitted system used by team aicyber. Then a new framework of estimating sentence vector will be introduced and evaluated.

2 The aicyber system

This section introduces the submitted system for team aicyber. The text data is first being processed by tweet tokenizer, emoticons are preserved as tokens. Bag-of-ngram feature is extracted and filtered by a TF-IDF (Salton, 1991) selection. Resulting feature dimension is around 3800, it is then reduced to 400 by truncated singular value decomposition (SVD) (Klema and Laub, 1980; Halko et al., 2009). This process is also known as Latent Semantic Analysis (LSA) or Vector Space Model (Turney and Pantel, 2010). Finally a Linear Discriminant Analysis (LDA) classifier (Hastie et al., 2009) is trained to classify the test data.

The SemEval 2016 training dataset which contains 3887 tweets are selected to train the TF-IDF, SVD and LDA. Development dataset is used for tuning parameters and develop-test dataset are used for local testing.

Task A adopted Macro-F1 measure as evaluation metric (Nakov et al., 2016):

$$F_1^{PN} = \frac{F_1^{Pos} + F_1^{Neg}}{2} \tag{1}$$

The Macro-F1 for Aicyber system measured on development, develop-test and 2016 Tweet set, are respectively 0.4514, 0.4787 and 0.4024.

It is obvious that this classification problem has not been satisfactorily answered. One possible reason is the unbalanced training data causes system bias towards positive classes. There are 2101 positive, 1292 neutral but only 494 negative tweets.

Another reason is the size of labeled training set is too small,with only 3887 tweets, which could hardly cover a reasonable amount of words. As a result, the bag-of-ngarm features learned from this training set, could not generalized well. This motivate us to seeking alternative feature representation of tweet, that is sentence vector.

3 Sentence vector

(Le and Mikolov, 2014) proposed sentence vector or paragraph vector (PV) which could learn the continuous distributed vector representation for

Proceedings of SemEval-2016, pages 120–125,
San Diego, California, June 16-17, 2016. ©2016 Association for Computational Linguistics

text of variable-length and achieved promising result on movie review texts. It is inspired by word2vec (Mikolov et al., 2013) embedding which captures rich notions of semantic relatedness and constitutionality between words. (Mesnil et al., 2014) shows the ensemble of sentence vector, RNN language model (Mikolov et al., 2010) and NB SVM (Wang and Manning, 2012) achieved new state-of-the-art result.

(Dai et al., 2015) extends the study and provide a more thorough comparison of PV to other task such as document modeling. It concluded PV can effectively be used for measuring semantic similarity between long pieces of texts.

However such approach assume testing data is known during learning of vector representation. Access of testing data during training may not allowed for certain machine learning task, and it is not practical for real application.

Thus we would like to introduce a new approach to estimate sentence vector or PV for variable-length of texts from word embeddings by using i-vector framework.

3.1 i-vector framework in speech domain

i-vector (Dehak et al., 2011) is one of the dominant approaches in speaker verification (SV) research in the recent years. It projects variable length speech utterances into a fixed-size low-dimensional vector, namely i-vector. Its development advanced from traditional techniques such as the Gaussian Mixture Models (GMMs) (Rose and Reynolds, 1990; Reynolds and Rose, 1995) , adapted GMMs (Reynolds et al., 2000), GMM supervectors (Campbell et al., 2006) and Joint Factor Analysis (Kenny et al., 2007).

To understand i-vector, firstly we need to understand the speech data, the speech data is a sequence of frames. At each frame a fixed-size feature vector is extracted, such as MFCC (Davis and Mermelstein, 1980), with addition of dynamics features such as ”delta” and ”delta-delta” (Furui, 1986). Thus one utterance could be viewed as a list of continuous-valued vectors as shown in Fig.1. A large amount of utterances are used to train the background GMM in i-vector framework. Secondly we need to learn the supervector. Its aim was to convert a spoken utterance with arbitrary duration to a fixed length vector. The supervector mentioned here is specific to the GMM supervector constructed by stacking the means of the mixture components. For example, a GMM with 2048 components built on 60 dimensional feature vectors results a 122880 (2048*60) dimensional supervector. A review of GMM and supervector is presented in (Kinnunen and Li, 2010).

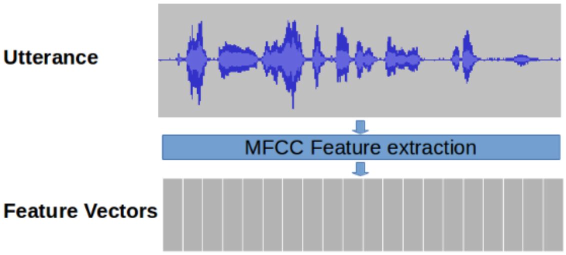

Figure 1: In i-vector framework of SV system, during data preprocessing, spoken utterance is presented as a list of feature vectors. A trained i-vector extractor could map utterances of various length into fixed-size vectors.

Given an utterance, its GMM supervector s can be represented as follows:

$$s = m + Tw \tag{2}$$

where m denotes Universal Background Gaussian Mixture Model’s (UBM) supervector, T is the total-variability matrix, which is used to represent the primary directions’ variation across a large amount of training data. The coefficients w of this total variability is known as identity vector or simply i-vector.

Extraction of this i-vector can be done as follows (Given a SV system built on F dimensional MFCC features and UBM with C Gaussian components):

$$w = (I + T^t\Sigma^{-1}NT)^{-1}T^t\Sigma^{-1}A \tag{3}$$

where I is $F \times F$ identity matrix, N is a $CF \times CF$ diagonal matrix whose diagonal blocks are N_cI(c=1,2,....C), and the supervector A is generated by the concatenation of the centralized first-order Baum-Welch statistics. Σ is the covariance matrix of the residual variability not captured by T. The i-vector’s dimension is usually 400, much lower than that of supervectors. This thus allows the use of various techniques that were not practical in high dimensional space. To give a completed review of i-vector is out of scope of this paper, interested individuals are strongly encouraged to read (Kenny et al., 2008; Dehak et al., 2011).

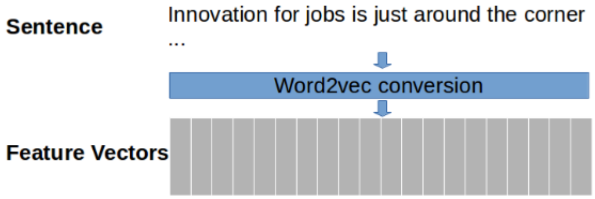

Figure 2: Data preprocessing of proposed NLP i-vector framework. Sentence is represented by its word embedding. A trained i-vector extractor could map sentence of arbitrary length into one fixed-size vector.

3.2 i-vector framework for Natural Language Processing task

(Shepstone et al., 2016) points out that *the central to computing the total variability matrix and i-vector extraction, is the computation of the posterior distribution for a latent variable conditioned on an observed feature sequence of an utterance.* In Natural Language Processing(NLP) when observations (words) could be represented as sequences of feature vectors, will the same methodology apply? This motivated us to bring i-vector from speech to NLP domain. Fig. 2 illustrates the fundamental principle of proposed i-vector framework for NLP task, where a sentence is represented through its word embedding during data preprocessing. Delta and delta-delta feature is added during training in order to capture the context information. Compare to Fig. 1 where spoken utterance is viewed as list of frame level MFCC feature vectors. The proposed i-vector framework replace MFCC features by word vectors, and trained using similar implementation as of speech data.

3.3 The implementation of i-vector framework

Many implementations of i-vector framework are developed recently, (Glembek et al., 2011) use standard GMM approach. (Snyder et al., 2015) incorporated time delayed deep neural network (TDNN) trained on speech recognition task into i-vector framework. A 50% relative improvement is obtained when TDNN instead of GMM is used to collect first-order Baum-Welch statistics. In this paper we use the light weighted conventional GMM approach for proof of concept purpose.

4 Experiment and evaluation

Training of i-vector system is in a completely unsupervised manner, it includes training of word2vec and training of i-vector extractor. Evaluation is done on IMDB similar to (Maas et al., 2011; Le and Mikolov, 2014; Mesnil et al., 2014) and SemEval 2016 Task4A.

4.1 Word2vec training

Word2vec training is done by using gensim [1]. Training dataset is selected from IMDB, it contains 25000 labeled training samples and 50000 unlabeled data, a total of 75000 movie reviews.

The training use a context window of 7, minimum word count of 10 and the resulting dimension of word vector is 20.

4.2 i-vector extractor training

Same data from word2vec training is used for i-vector extractor training. As illustrated in Fig 2, the data preprocessing involved word-to-vector conversion. Words that not appear in the word2vec model are ignored. Each review is treated as one sentence. The sentence is saved as list of word vectors, can be viewed as a $M \times 20$ matrix. Where M denotes number of words in that sentence.

The proposed i-vector extractor training system[2] is developed using the Kaldi Speech Recognition Toolkit (Povey et al., 2011)[3]. Feature used is a 60 dimensional features consisting of the 20 dimensional word vector and its delta, double delta. Mean and variance normalization is not applied. During training, a universal background model (UBM) with 2048 Gaussian mixture components is trained on 64000 sentences. Each Gaussian in UBM has a full covariance matrix. After UBM is trained, the total variability matrix is similar trained with all the 75000 sentences. Learned i-vector extractor is then used to estimate vectors for IMDB and SemEval 2016 dataset. The output dimension of i-vector is 200. Note that, test data of IMDB and all data in SemEval 2016 are not observed during training.

[1] http://radimrehurek.com/gensim/

[2] https://github.com/StevenLOL/aicyber_semeval_2016_ivector

[3] https://github.com/kaldi-asr/kaldi

System	Training Data	Accuracy (%)
State-of-the-art	train+test+unlabeled	**92.57**
Sentence Vectors	train+test+unlabeled	88.73
Aicyber's system	train	88.38
i-vector	train+unlabeled	**87.52**
i-vector+VSM	train+unlabeled	**89.94**

Table 1: Proposed i-vector based sentence vector evaluated on the IMDB dataset. It didn't beat the state-of-the-art, an ensemble of three sub systems.

4.3 Evaluation of proposed framework

The i-vector framework is first evaluated on the IMDB dataset then on the SemEval 2016 dataset.

4.3.1 Evaluation on IMDB

Evaluation metric is accuracy measured on IMDB database. Table 1 shows the performance of different systems. The current state-of-the-art system is an ensemble of RNN language model, sentence vectors and NB SVM, achieved 92.57% (Mesnil et al., 2014) testing accuracy. Sentence vector system is one of sub-system used in ensemble and achieved 88.73% accuracy alone. Aicyber's system is the same system mentioned in Section 2, a VSM approach, it obtained 88.38%. To make a fair comparison same type of classifier, LDA is used to train and classify i-vector system, a 87.52% accuracy is reported. Concatenation of i-vector and vector from VSM a 89.94% accuracy can be achieved.

4.3.2 Evaluation on SemEval 2016

Evaluation metric for SemEval 2016 task is Macro-F1 introduced in Equation 1. During evaluation period we validate the performance on development and develop-test dataset. Results as shown in Table 2 indicate i-vector system is worse than our baseline system. So we only submitted the baseline system.

4.4 Discussion

Judging from the IMDB evaluation, the idea of i-vector from speech domain is successfully applied for NLP task. Though it didn't beat the state-of-the-art, it is well-known that basic machine learning techniques can yield strong baselines (Wang and Manning, 2012) on this dataset.

Performance dropped in SemEval 2016 could due to data mis-match, the word2vec and i-vector extractor is trained on the movie review texts which are much longer, more formal than tweets and are of different vocabulary.

System	Aicyber's system	i-vector system
Development (Macro-F1)	0.4514	0.3732
Development-test (Macro-F1)	0.4787	0.3814

Table 2: Proposed i-vector based sentence vector evaluated on the SemEval 2016 dataset. Aicyber's system is VSM approach and trained on 3887 tweets. The i-vector extractor learned from IMDB dataset and apply on SemEval data. The results show that the i-vector system could not beat our submitted system.

Further improvement can be made in the following aspects.

1. The word-to-vector conversion is done via word2vec, using of Glove word vectors (Pennington et al., 2014), either alone or as concatenation of two type of word vectors for system training is worth to explored.

2. Parameters in word2vec, GMM and i-vector training is not yet optimized to the task. For example, word vector is set to 20 dimensions which is much smaller than 300 dimensional Google word vector. Finding the best parameters will bring more insight of semantical meaning of sentence vector.

3. In this paper, the implementation of i-vector framework is a GMM based approach. Incorporating deep neural network (Snyder et al., 2015) or a Convolutional Neural Network (Kalchbrenner et al., 2014) for NLP task is worth to investigated.

5 Conclusion

This paper had presented a vector space model approach for team aicyber and a new idea of estimating sentence vector. Proposed i-vector framework had evaluated on SemEval 2016 as well as IMDB dataset. Result shows that the i-vector based sentence vector is an alternative approach to present sentence.

References

W. M. Campbell, D. E. Sturim, and D. A. Reynolds. 2006. Support vector machines using gmm supervectors for speaker verification. *IEEE Signal Processing Letters*, 13(5):308 – 311.

A. M. Dai, C. Olah, and Q. V. Le. 2015. Document Embedding with Paragraph Vectors. *ArXiv e-prints*, July.

S. B. Davis and Paul Mermelstein. 1980. Mermelstein, p.: Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. ieee trans. acoust speech signal processing 28(4), 357-366. *IEEE Transactions on Acoustics Speech and Signal Processing*, 28(4):357–366.

N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet. 2011. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio Speech and Language Processing*, 19(4):788–798.

Sadaoki Furui. 1986. Speaker-independent isolated word recognition based on emphasized spectral dynamics. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP*, pages 1991–1994.

O. Glembek, L. Burget, P. Matejka, and M. Karafiat. 2011. Simplification and optimization of i-vector extraction. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 4516 – 4519.

N. Halko, P. G. Martinsson, and J. A. Tropp. 2009. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *Siam Review*, 53(2):217–288.

T. Hastie, R. J. Tibshirani, and J. H. Friedman. 2009. The elements of statistical learning: Springer. *Chemistry International*, (6):91–108.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *Eprint Arxiv*, 1.

P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel. 2007. Joint factor analysis versus eigenchannels in speaker recognition. *IEEE Transactions on Audio Speech and Language Processing*, 15(4):1435–1447.

P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel. 2008. A study of inter-speaker variability in speaker verification. In *IEEE Trans. Audio, Speech and Language Processing*, pages 980 – 988.

Tomi Kinnunen and Haizhou Li. 2010. An overview of text-independent speaker recognition: From features to supervectors. *Speech Communication*, 52(1):1240.

V. Klema and A. J. Laub. 1980. The singular value decomposition: Its computation and some applications. *IEEE Transactions on Automatic Control*, 25(2):164–176.

Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. *Eprint Arxiv*, 4:1188–1196.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, pages 142–150.

Grégoire Mesnil, Tomas Mikolov, Marc'Aurelio Ranzato, and Yoshua Bengio. 2014. Ensemble of generative and discriminative techniques for sentiment analysis of movie reviews. *arXiv preprint arXiv:1412.5335*.

Tomas Mikolov, Martin Karafit, Lukas Burget, Jan Cernock, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. *Interspeech*, pages 1045–1048.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26:3111–3119.

Preslav Nakov, Alan Ritter, Sara Rosenthal, Veselin Stoyanov, and Fabrizio Sebastiani. 2016. SemEval-2016 task 4: Sentiment analysis in Twitter. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California, June. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Luk Burget, Ondej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlek, Yanmin Qian, and Petr Schwarz. 2011. The kaldi speech recognition toolkit. *Idiap*.

D. A. Reynolds and R. C. Rose. 1995. Robust text-independent speaker identification using gaussian mixture speaker models. *IEEE Transactions on Speech and Audio Processing*, 3(1):72–83.

Douglas A. Reynolds, Thomas F. Quatieri, and Robert B. Dunn. 2000. Speaker verification using adapted gaussian mixture models. *Digital Signal Processing*, 10(1-3):19–41.

R. C. Rose and D. A. Reynolds. 1990. Text independent speaker identification using automatic acoustic segmentation. In *International Conference on Acoustics, Speech, and Signal Processing, Icassp*, pages 293 – 296.

Gerard Salton. 1991. Developments in automatic text retrieval. *Science*, 253(5023):974–980.

Sven Shepstone, Kong Aik Lee, Haizhou Li, Zheng-Hua Tan, and Soren Holdt Jensen. 2016. Total variability modeling using source-specific priors.

David Snyder, Daniel Garcia-Romero, and Daniel Povey. 2015. Time delay deep neural network-based universal background models for speaker recognition. ASRU.

Peter D. Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37(4):141–188.

Sida Wang and Christopher D. Manning. 2012. Baselines and bigrams: simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, pages 90–94.