Semi-supervised Learning of Dialogue Acts Using Sentence Similarity Based on Word Embeddings

Xiaohao Yang, Jia Liu
Department of Electronic Engineering, Tsinghua University
Beijing 100084, China
Email: yangxiaohao@gmail.com

Zhenfeng Chen, Weilan Wu
University of Chinese Academy of Sciences
Beijing 100190, China
Email: chenzfnolove@foxmail.com

***Abstract*—This paper describes a methodology for semi-supervised learning of dialogue acts using the similarity between sentences. We suppose that the dialogue sentences with the same dialogue act are more similar in terms of semantic and syntactic information. However, previous work on sentence similarity mainly modeled a sentence as bag-of-words and then compared different groups of words using corpus-based or knowledge-based measurements of word semantic similarity. Novelly, we present a vector-space sentence representation, composed of word embeddings, that is, the related word distributed representations, and these word embeddings are organised in a sentence syntactic structure. Given the vectors of the dialogue sentences, a distance measurement can be well-defined to compute the similarity between them. Finally, a seeded k-means clustering algorithm is implemented to classify the dialogue sentences into several categories corresponding to particular dialogue acts. This constitutes the semi-supervised nature of the approach, which aims to ameliorate the reliance of the availability of annotated corpora. Experiments with Switchboard Dialog Act corpus show that classification accuracy is improved by 14%, compared to the state-of-art methods based on Support Vector Machine.**

keywords-word embeddings; sentence similarity; dialog acts; seeded k-means

I. Introduction

An important task in spoken dialog systems is dialog act modeling which determines the dialogue acts of users' utterances since they capture the action or the communication goal underlying each utterance[1]. In practice, dialogue act modeling is a classification process based on creating taxonomies[1][2].

The approaches for learning the dialog act model from a corpus fall into two major aspects: supervised learning and unsupervised learning. Supervised learning requires a manually labeled corpus to train a model, while unsupervised learning employs several machine learning methods which rely solely on the structure of the unlabeled data. As for the supervised learning of dialog acts, a rich literature has shown success by a supervised classifier Support Vector Machine (SVM) with a variety of lexical input features[3][4][5]. However, supervised learning faces two significant challenges. Firstly, manual annotation is an expensive and time-consuming task, which forms a major bottleneck in annotating for new corpora of different domains. Secondly, the tagging scheme itself is often not well-defined even though different annotators reach an agreement when tagging based on a standard method[1]. On the contrary, unsupervised learning does not rely on manual labels, and clusters the corpus into several partitions according to the inner structure of the corpus though a fully data-driven way. Recent study suggests that unsupervised approaches have shown promising performance for dialog act classification[6][7]. However, unsupervised learning suffers from limitations that the initial partitions are ill-defined and the resulting clusters are undeclared[8].

To address these drawbacks, this paper presents a novel approach toward semi-supervised dialog act classification with partial labeled data. The approach employs sentence similarity as the distance measurement in the unlabeled corpus data. Most of the existing methods for computing sentence similarity apply to long texts in which word occurrence or co-occurrence can be an important statistical information, thus Latent Semantic Analysis (LSA) method can be implemented to represent the text with sparse vectors[9]. However, in consideration of the fact that the dialog sentences are mostly short texts (no more than 15 words), word occurrence can hardly capture the exact meaning of the dialog sentences. Meanwhile, the syntactic information at the sentence level is highly relevant in the computation of similarity for short sentences. Therefore, the meaning of a dialog sentence is composed of not only the semantic meaning of the individual words, but also the syntactic information about the organisation of the words. In this paper we describe a new method to evaluate the similarity between dialog sentences by taking into account syntactic and semantic information. Given the similarity of all the sentence pairs in corpus as a distance measurement, we can implement a seeded k-means clustering algorithm using partial labeled data to find the initial centroids. Finally, we conduct a quantitative evaluation with respect to manual labels, compared with a state-of-the-art supervised SVM baseline[3][10] and an unsupervised baseline[8].

The outline of the whole procedure is showed in Figure 1. The paper starts with word similarity based on word embeddings which are trained with recent neural networks[11][12] in Section 2. Section 3 describes vector space representations of sentences combined with word-level semantic and sentence-level syntactic information and a novel approach to computing the similarity between sentences. Once the sentences are represented by vectors with the same dimension, a seeded k-means clustering is consequently implemented in Section

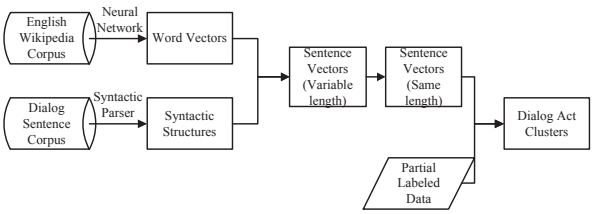

Fig. 1: The whole procedure of semi-supervised dialog act learning.

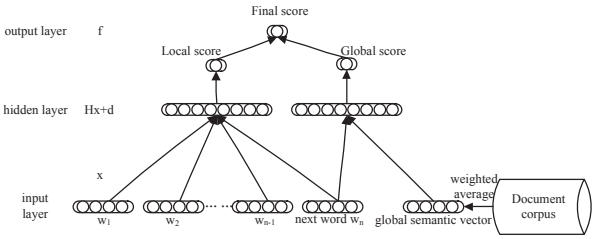

Fig. 2: The neural network structure of word embedding training[12]

4. Section 5 presents a group of experiments on Switchboard Dialog Act corpus[13] and the results are discussed in Section 6. The paper concludes in Section 7.

II. Word similarity

Most of the effective methods compute the similarity between words according to WordNet[14], using its hierarchical structure and a variety of glosses associated with each term. Recent statistical methods also use WordNet as a lexical database and the incorporation of statistics is adapted to different domains[15]. With the explosive increasing of knowledge and data, a predefined lexical database may not update as well and the relevant human judges are sometimes less accurate than statistical regularities learned from a large corpus.

In this work, we employ recent word embeddings of words to measure the similarity between them[11]. We use a three-layer neural network model learning both local and global context information via a joint training objective[12]. It induces word embeddings that capture the semantics of words, while still keeping their syntactic information. These properties lay a foundation for the structural representations of sentences by these embeddings in later sections.

Figure 2 shows the structure of the neural network which learns word embeddings[12].The input word representation is traditional one-hot representation in form of $(0, 0, 1, 0, 0, 0, 0, 0)$ and each word takes a symbolic ID. The sparse representation makes every two words isolated, even those words with similar semantics. The goal is to learn useful word representations so both the local and global information should be concerned as the input. Then a pair-wise method is used for training the model. Given a word sequence s and document d which contains s, we compute the score $g(s, d)$ and $g(s^\omega, d)$ where s^ω is s with the last word replaced by another word ω. Consequently the three-layer neural network is trained, with a goal that the score $g(s, d)$ is bigger than $g(s^\omega, d)$ by a margin of 1, or minimize the following goal function:

$$\sum_{\omega \in V} \max(0, 1 - g(s, d) + g(s^\omega, d)) \tag{1}$$

where V represents the whole vocabulary.

With this pair-wise ranking approach, good word embeddings can be produced. However, the function only concerns the local information near the target word while the whole document also contributes the global context information to the word. In this paper two scoring components are used[12]. The local score places extra emphasis on word order and syntactic information, while the global score gives more semantic and topic information. Therefore we choose the neural network in the figure[12] to train word embeddings. The output word representations look like $(0.791, -0.177, -0.105, 0.209, -0.542, ...)$, whose dimensions are normally 50 or 100. The exciting thing is that the similarity between words can be briefly measured by Euclidean or cosine distance between the word embeddings. Recent study shows that the semantic and syntactic relationship between words can be also derived from the simple offset between vectors[16]. For example , *King - Man + Woman* results in a vector very similar to *Queen*. We suppose that a sentence vector can be composed by the word emdeddings, along with a syntactic structure. The following Sections will describe it in detail.

During the actual implementation of training, we use the English Wikipedia corpus[12] with a vocabulary of $100,232$ words. The dimension of the output embeddings is 50, due to the expectation that the compositional sentence vectors are located in a relatively low dimensional space.

Given the representations of words, the similarity between them is consequently measured by the distance between the vectors such as Euclidean distance and cosine distance. We can spontaneously conduct a traditional word-by-word comparison to compute the similarity between two sentences. However, we suggest a sentence-level representation which captures both semantic and syntactic information and similarity is computed on those sentence-level vectors composed of word embeddings.

III. Sentence similarity

A. Sentence representation

To quantitatively measure the similarity between sentences, the first step is to represent a sentence by a numerical vector. The intuitive approach is simply adding up the vectors of words in a sentence. For example, if we denote a sentence as T with a word set $\{T\}$. We already get the word ebeddings for each word ω as X_ω, then the whole sentence representation X_T is as follows:

$$X_T = \sum_{\omega \in \{T\}} X_\omega \tag{2}$$

However, this linear adding method, which treats a sentence as a bag of words, hardly capture the syntactic information how the words are organised. For example, we consider two

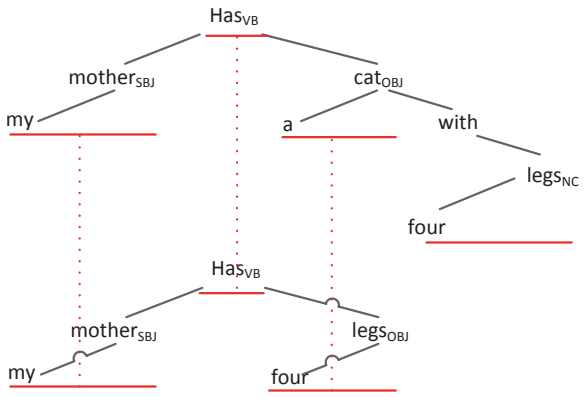

Fig. 3: Parse trees of the sentences: "my mother has a cat with four legs" and "my mother has four legs". The phrases which have the same syntactic function are compared, which are underlined with red lines.

sentences, *my mother has a cat with four legs* and *my mother has four legs*. Using bag-of-words method, the similarity between the two sentences is quite high while they are not actually very similar. In this paper we explore a sentence representation composed by word embeddings in a structural way.

In the example mentioned above, the method would work as follows: At first, we carry out a syntactic analysis of the sentences, obtaining the parse trees in Figure 3 by the system of Nugues and Johansson[17], which concerns both semantic role labeling and syntactic parsing. Once the syntactic trees have been extracted, we generate a higher dimensional vector than a word embedding, following the formula below:

$$X_T = \{t_1, X_{p_{t_1}}, t_2, X_{p_{t_2}}, ..., t_n, X_{p_{t_n}}\} \tag{3}$$

We assume that sentence T is made of n phrases and their syntactic functions are labeled as $t_i (i = 1, ..., n)$. Word embeddings in the same type of syntactic function are added together to induce a phrase vector which has the same dimension with word embeddings. This constitutes the vector space representation of a sentence. We use these sentence-level vectors to compute the similarity between sentences.

B. Similarity method between sentences

Even though the vectors of sentences are given, we cannot implement the common distance measurement between numerical vectors directly. Due to the structural feature of sentences, the same syntactic function may be located in different positions in different sentences. Before comparing two sentences, we firstly carry out an alignment of the same syntactic function and then the cosine distance is calculated between the phrase vectors that have the same syntactic function. If one sentence has a syntactic function not shared by the other, a penalization factor is produced by the percentage of the extra phrase in the longer sentence[18]. Then, the similarity between two sentences would be computed as follows:

$$sim(X_{T_1}, X_{T_2}) = max\{\frac{1}{m}\sum_{i=1}^{m}|cos(X_{p_{1i}}, X_{p_{2i}})| - \frac{k}{n}, 0\} \tag{4}$$

where p_{1i} and p_{2i} are in the same type of syntactic function, and the penalization factor is defined as the percentage of extra phrases. Here k means the number of the extra phrases and n means the number of the total phrases in the longer sentence. According to the formula (4), the range of similarity is in range of [0,1], in which 1 means the two sentences are exactly the same and 0 means there is hardly similarity between them or there is too many extra information in one of the sentence pairs.

IV. Semi-supervised dialog sentences clustering

The similarity calculated by formula (4) is used as the distance metric in a k-means clustering algorithm. However, the clustering process faces two challenges. First of all, the input vectors have to be in the same dimensional vector-space but the vectors produced by formula (3) are non-uniformed. Secondly, the initial centroids are defined randomly without the information of the k. To address these problems, we create a *similarity-vector* for a sentence. The idea relies on calculating the similarities between the target sentence and all the sentences in the data set, and we represent the target sentence with a vector composed of these similarities. The similarity is obviously 1 compared to itself. The resulting vectors have the same dimension with the number of the sentences in data set and the value of each dimension is normalised to $[0, 1]$ according to the formula (4). The uniformed sentence-level vector is formed as follows:

$$\begin{aligned}&\text{Norm}X_{T_i} = \\ &\{sim(X_{T_i}, X_{T_1}), sim(X_{T_i}, X_{T_2}), ..., sim(X_{T_i}, X_{T_n})\}\end{aligned} \tag{5}$$

where n means the number of the sentences in the data set and $i = 1, ..., n$.

Finally a seeded k-means clustering is carried out with the initial centroids decided by partial labeled data. The entire semi-supervised dialog act clustering algorithm is summarised as Algorithm 1.

V. Experimental Evaluation

A. Data set

The Switchboard Dialog Act corpus (SwDA)[13] is used for dialog act classification. SwDA has more than $200,000$ sentences with 42 dialog acts, among which 10 simplified dialog acts are chosen for our experiment, along with their relevant sentences about $50,000$. As for the semi-supervised learning, we choose 10 sentences for each dialog act type as our partial labeled data to find the initial centroids.

B. Evaluation Criteria

We calculate the precision, recall and F scores for each class of the dialog acts. The total classification accuracy (CA) is also calculated to compared with state-of-the-art supervised

Algorithm 1 seeded k-means clustering

Input: Set of sentence vectors $X = \{x_1, ..., x_N\}, x_i \in R^n$ in formula (5), number of clusters K, set $S = \cup_{h=1}^{K} S_h$ as initial seeds
Output: Disjoint K clusters $\{X_1, ...X_K\}$ such that k-means objective function is optimized
Method
initialize: $\mu_h^{(0)} \leftarrow \frac{1}{|S_h|} \sum_{x \in S_h} x$, for $h = 1, ..., K; t \leftarrow 0$
repeat
1: **assign cluster**: Assign each x to the cluster h^*, for $h^* = \arg\min_h ||x - \mu_h^t||^2$
2. **estimate means**: $\mu_h^{t+1} \leftarrow \frac{1}{|X_h^{t+1}|} \sum_{x \in X_h^{t+1}} x$
3. $t \leftarrow t + 1$
until criterion function converge

baselines one of which is Support Vector Machine (SVM) method[3], the other employing a multi-classifier approach to improve the performance[10]. Moreover, we compare our semi-supervised method with unsupervised baselines in recent study[8].

C. Experimental results

Table 1 shows the comparison between the state-of-the-art supervised approach[10] and our semi-supervised approach for the performance of each class. Table 2 presents several systems for dialog act classification, including two supervised baselines, unsupervised baseline and our system. In order to examine the effectiveness of our method of computing sentence similarity, we also plug our sentence representations into the existing unsupervised system to see if it can improve the performance..

TABLE I: PRECISION, RECALL AND F SCORES FOR EACH CLASS USING SUPERVISED(SU) OR SEMI-SUPERVISED(SE) METHODS. VALUES ARE PERCENTAGES.

Tag	Precision		Recall		F-score	
Dialog act	SE	SU	SE	SU	SE	SU
Statement(51.1%)	75.5	77.3	71.2	72.1	73.3	74.6
opinion(16.32%)	65.3	67.7	58.8	60.0	61.8	63.6
Agree/Accept(12.89%)	78.0	80.3	75.3	75.3	76.6	77.7
Appreciation(5.14%)	77.1	78.0	80.3	80.7	78.7	79.3
Yes-No-Ques(4.11%)	88.1	88.3	**90.2**	89.1	**89.1**	88.7
Yes answers(3.51%)	**91.4**	91.2	**93.5**	92.7	**92.4**	92.0
Closing(2.78%)	**60.8**	60.7	**62.3**	61.4	**61.6**	61.0
Wh-Question(1.52%)	**88.7**	85.1	**71.5**	67.2	**79.2**	75.1
No answers(1.43%)	**94.8**	91.1	**91.9**	90.8	**93.4**	90.9
Response Ack(1.3%)	**74.2**	73.4	**75.6**	73.5	**74.9**	73.5

VI. DISCUSSION

A strength of semi-supervised approaches is that they rely on few manually tagging schemes and reflect the inner structure of the corpus in a data-driven way. However, a fully unsupervised method suffers from the undefined initial partitions. We leverage a small part of the labeled data(0.2%) to carry out the seeded k-means which results in better performance than supervised SVM classifier and unsupervised k-means method.

TABLE II: CLASSIFICATION ACCURACY FOR SEVERAL METHODS.

Metric	%CA
SVM[3]	71.55
Multi-classifier[10]	78.85
unsupervised+WordNet similarity[8]	76.31
unsupervised+word embedding similarity	**78.62**
semi-supervised+word embedding similarity	**81.57**

The results in Table 1 imply that when training data is sufficient, the state-of-art supervised method[3] outperforms the semi-supervised method. However, as the labeled data decreases, semi-supervised method proves an advantage. Thus our approach holds great promise for dialog act classification when sufficient labeled data is hard to obtain.

Moreover, results in Table 2 suggest that an unsupervised method still suffers degradation in classification performance compared to our semi-supervised method since the initial partitions are not well-defined. Noticing that we present a novel approach computing similarity between sentences by their semantic and syntactic information and the semantic information is captured by word embeddings or phrase embeddings, the performance related to different methods of sentence similarity is experimented on the same corpus. With the same k-means clustering algorithm, the results show that our measurement of similarity is well-defined for the clustering, outperforming the recent technology of sentence similarity[8].

VII. CONCLUSIONS

The paper mainly presents two contributions compared to the previous related work. One contribution is that we propose a new method for computing sentence similarity based on word embeddings and sentence syntax. Our method gives a superior numerical representation of a sentence and it improves the performance of the existing k-means approach as a new measurement of distance. The other contribution is that we present a novel dialogue act learning method which classifies dialog acts in a semi-supervised fashion by seeded k-means. The ill-defined initial centroids in k-means approach are well determined by the partial labeled data. Although the technique solely utilizes few manual labels for the initial part, the results are encouraging when compared to manually applied dialogue act labels. The experimental results show that our semi-supervised method performs substantially better than state-of-art supervised classification and unsupervised clustering.

REFERENCES

[1] A. Stolcke, K. Ries, N. Coccaro, E. Shriberg, R. Bates, D. Jurafsky, P. Taylor, R. Martin, C. Van Ess-Dykema, and M. Meteer, "Dialogue

act modeling for automatic tagging and recognition of conversational speech," *Computational linguistics*, vol. 26, no. 3, pp. 339–373, 2000.

[2] H. Bunt and Y. Girard, "Designing an open, multidimensional dialogue act taxonomy," in *Proceedings of the 9th Workshop on the Semantics and Pragmatics of Dialogue (DIALOR 2005)*. Citeseer, 2005, pp. 37–44.

[3] D. Surendran and G.-A. Levow, "Dialog act tagging with support vector machines and hidden markov models." in *INTERSPEECH*, 2006.

[4] Y. Liu, "Using svm and error-correcting codes for multiclass dialog act classification in meeting corpus." in *INTERSPEECH*, 2006.

[5] J. D. OShea, Z. A. Bandar, and K. A. Crockett, "Optimizing features for dialogue act classification," in *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*. IEEE, 2013, pp. 474–479.

[6] N. Crook, R. Granell, and S. Pulman, "Unsupervised classification of dialogue acts using a dirichlet process mixture model," in *Proceedings of the SIGDIAL 2009 Conference: The 10th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Association for Computational Linguistics, 2009, pp. 341–348.

[7] A. Ritter, C. Cherry, and B. Dolan, "Unsupervised modeling of twitter conversations," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Stroudsburg, PA, USA, 2010, pp. 172–180.

[8] A. Ezen-Can and K. E. Boyer, "Unsupervised classification of student dialogue acts with query-likelihood clustering," in *International Conference on Educational Data Mining*, 2013, pp. 20–27.

[9] T. K. Landauer and S. T. Dumais, "A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge." *Psychological review*, vol. 104, no. 2, p. 211, 1997.

[10] J. OShea, Z. Bandar, and K. Crockett, "A multi-classifier approach to dialogue act classification using function words," in *Transactions on Computational Collective Intelligence VII*. Springer, 2012, pp. 119–143.

[11] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems*, 2013, pp. 3111–3119.

[12] E. H. Huang, R. Socher, C. D. Manning, and A. Y. Ng, "Improving word representations via global context and multiple word prototypes," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, 2012, pp. 873–882.

[13] D. Jurafsky, E. Shriberg, and D. Biasca, "Switchboard swbd-damsl shallow-discourse-function annotation coders manual," *Institute of Cognitive Science Technical Report*, pp. 97–102, 1997.

[14] T. Pedersen, S. Patwardhan, and J. Michelizzi, "Wordnet:: Similarity: measuring the relatedness of concepts," in *Demonstration Papers at HLT-NAACL 2004*. Association for Computational Linguistics, 2004, pp. 38–41.

[15] Y. Li, D. McLean, Z. A. Bandar, J. D. O'shea, and K. Crockett, "Sentence similarity based on semantic nets and corpus statistics," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 18, no. 8, pp. 1138–1150, 2006.

[16] T. Mikolov, W.-t. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," in *Proceedings of NAACL-HLT*, 2013, pp. 746–751.

[17] R. Johansson and P. Nugues, "Dependency-based syntactic-semantic analysis with propbank and nombank," in *Proceedings of the Twelfth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, 2008, pp. 183–187.

[18] J. Oliva, J. I. Serrano, M. D. del Castillo, and Á. Iglesias, "Symss: A syntax-based measure for short-text semantic similarity," *Data & Knowledge Engineering*, vol. 70, no. 4, pp. 390–405, 2011.