Dialog Act Classification Using Utterance Embeddings

Lautaro Quiroz 10849963
Roger Wechsler 10850007
David Woudenberg 10069143

Abstract

In this work, we experiment with a new feature representation for dialog act tagging by learning distributional embeddings for utterances. We train a distributional semantic model that allows us to infer vector representations for entire utterances in an unsupervised fashion. These representations are used to train several classifiers in order to tag utterances in the Switchboard corpus. We show that these utterance embeddings can be used for dialog act tagging. Our method, however, is outperformed by a simple bag-of-words baseline.

1 Introduction

Discourse structure analysis is fundamental for understanding spontaneous dialogs and developing human-computer dialog systems. An essential part of discourse structure analysis is the identification of dialog act classes (e. g. *questions*, *self-talks*, *statements*, *backchannels*). As defined by Austin (1962), dialog acts present linguistic abstractions of the ilocutionary force of speech acts and model the communicative intention of an utterance in a conversation. There are several tasks that require utterances to be tagged with dialog acts. Examples of these include speech recognition, speech synthesis, summarization and, of course, human-machine dialog systems. The correct identification of dialog act tags for utterances is thus an important research topic.

Table 1 shows an example of dialog acts from the Switchboard corpus we are trying to classify. The table already gives an idea that some of the 42 dialog acts might have a rather closed set of possible realisations (e.g. the classes *Agree* or *Acknowledge*) whereas classes like *Statement* can contain utterances of almost any content. Although the individual words in an utterance are important cues, we argue that the ilocutionary force, and thus the dialog act tag, of an utterance is derived from: 1) the *words* in that utterance, 2) their *composition* and 3) the *context* of the dialog as a whole.

Tag	Speaker / Utterance
Wh-Question	A: how old is your daughter?
Statement-non-opinion	B: she's three.
Summarize	A: oh , so she's a little one.
Agree	B: yes.
Acknowledge	A: yeah.
Statement-non-opinion	B: she's, she's little.

Table 1: SWDA dialog excerpt.

For the purpose of this work, we will focus on capturing the first two of these three aspects. We extract feature representations for complete, but isolated, utterances in an unsupervised fashion. We therefore build a distributional semantic model that learns vector representations for entire utterances and then use these vector features as inputs for different machine learning classifiers, expecting that the embeddings can model the meaning of an entire utterance. Several techniques can be used for mapping text units to a high-dimensional real value space. Utterance embeddings have the attractive property of representing an entire textual sequence as a vector while taking word order into account, as opposed to the classical *Bag-of-words* approach in which word order is not preserved and in which resulting vectors show only little semantic relations. We expect this additional information to play an important role in the classification task. In order to infer those embeddings we use the *paragraph2vec* framework recently introduced by Le and Mikolov (2014), which is based on the earlier word embedding models by Mikolov et al. (2013).

For the actual dialog act tagging we treat the problem as a multi-class classification task and classify utterances both in isolation as well as in the context of the previous utterances. We evaluate the tagging accuracy and compare different

models. Besides the results provided by research from previous work, we compare the performance of our approach against a simple baseline that uses a bag-of-words (BOW) representation for each utterance.

We will test a total of three hypotheses in this report.

Hypothesis 1 Utterance embeddings can be used for dailog act tagging.

Hypothesis 2 Classifiers using utterance embeddings outperform a Bag-of-Words baseline.

Hypothesis 3 Using additional data in the unsupervised training of utterance embeddings increases the accuracy of the classifiers.

The outline of this report is set as follows: In Section 2 we present relevant approaches that aim at classifying dialog acts and briefly describe their main characteristics and results. In Section 3 we describe the concept of utterance embeddings and their training in more detail. Section 4 specifies the datasets and the details of our classification pipeline. The results of the experiments are presented in Section 5. Finally, Section 6 includes conclusions drawn from this work as well as issues and future work.

2 Related Work

Several approaches have been proposed for classifying dialog acts. Most of them rely on supervised trained models and use hand-crafted features extracted for all utterances. Some recent work shows that using distributional representations for dialog act classification outperforms these methods. We briefly present some of the most relevant work in this section.

The authors of Stolcke et al. (2000) predict dialog acts by modeling a conversation as a Hidden Markov Model (HMM). A sequence of dialog acts is represented as a *discourse model* where the probability of the next dialog act depends on the *n* previous dialog acts. They integrate this model with a *language model* for each separate dialog act, which computes the probability for a certain dialog act tag given the occurrence of all *word n-grams* in that utterance. Stolcke et al. (2000) also train models on the actual speech signals, where the 'language model' is trained on prosodic and acoustic evidence. When considering the models trained on the dialog transcripts we can see that in this an utterance is represented as a bag of n-grams. This might not capture the entire composition of that utterance. We aim to train a single model to directly find the dialog act of an utterance given some representation of that utterance.

In Kalchbrenner and Blunsom (2013) a Recurrent Convolutional Neural Network (RCNN) is trained in a supervised manner on a dialog corpus, which achieves state of the art results on the dialog act tagging task. The RCNN learns an integrated *discourse model* and a *sentence model* from a specific corpus, where the utterance representation is derived from individual word vectors, which are chosen randomly. We feel that this representation can become a lot richer if it is learned as in Le and Mikolov (2014), where word vectors have some distributional meaning. Another strength of the approach of Le and Mikolov (2014) is that it is possible to train these utterance embeddings in an unsupervised manner, making it possible to include additional, possibly unannotated, corpora.

An investigation on the contribution of distributional semantic information to the dialog act tagging task was conducted in Milajevs and Purver (2014). It was found that such information did improve tagging when compared to simple bag-of-words approaches. However only very simple distributional representations were investigated in this work. Words were represented as vectors of their co-occurrence counts. The space of these vectors was reduced using Singular Value Decomposition (SVD) to obtain a denser representation. Utterances were then represented as pointwise multiplications or additions of these vectors, which implicates the loss of any compositional information. The work of Milajevs and Purver (2014) was not able to outperform the earlier work on dialog act tagging presented before. We believe that with richer representations of utterances, where we also incorporate composition, we can improve on this performance.

We will attempt to find a model that directly models the probability of dialog tags given a representation of an utterance. Unlike Stolcke et al. (2000) who learns a separate model for the dialog and for each dialog act. We will adopt techniques proposed by Le and Mikolov (2014), which we will explain in the next section, to capture the words and composition of an utterance. Just like Milajevs and Purver (2014) we will use these rich representations to train classifiers. Note that we

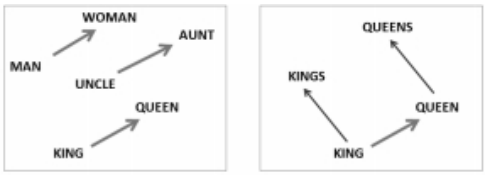

Figure 1: Word2vec semantic relations.

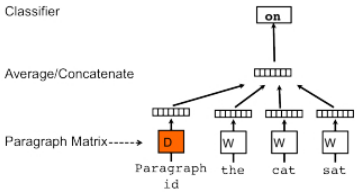

Figure 2: Paragraph vector PV-DM architecture.

will only model context by concatenating utterance representations and leave a more informative approach to modeling context for future work.

3 Utterance Embeddings

To represent utterances as dense distributional vectors we will use techniques from Le and Mikolov (2014). Their work will be explained exhaustively in this section, we start with presenting word embeddings as proposed by Mikolov et al. (2013) and then extend this idea in order to find embeddings for a sequence of words of an arbitrary length.

3.1 Word Embeddings

Neural networks for distributional semantics have gained relevant importance in the past years since the publication of the first neural network word embedding models. One of the main reasons for this was the discovery that dense high-dimensional vector representations of words are able to capture semantic relations between them (2013). Figure 1 shows a typical example where simple vector addition and subtraction lead to analogies such as: *Woman* is to *Man* what *Aunt* is to *Uncle*.

In distributional semantic models, word embeddings are learned by predicting words within a context of other words. Based on the assumption that similar words occur in similar contexts, these models are capable of successfully representing words in high-dimensional vectors.

Originally, these approaches had two main architectures, know as *Continuous-bag-of-words* and *Skip-gram*. In the first case, the neural networks were optimized to predict the next word given its context, whereas in the latter, the context is predicted given a certain word. Due to word co-occurrences, these models are able to effectively capture the semantic representation of the words.

3.2 Utterance embeddings

The approach from Mikolov et al. (2013) is no longer feasible for word sequences such as entire utterances, as the vocabulary of all possible utterances is infinitely large, because in theory we could construct utterances of any length. In order to overcome this sparsity problem, a slightly different framework called *paragraph2vec*[1] for learning representations of entire word sequences has been proposed by Le and Mikolov (2014), in which vectors are learned for a sequence from the words within the sequence. For our case, the model trains utterance embeddings as well as word embeddings.

During training, two structures are maintained: one for word, and one for utterance representations. The word structure is shared across the entire model, but an utterance embedding is trained for each single utterance individually. The training task is the same as before: Given a certain window, the model is optimized to predict the missing word. However this time, the context representation is constructed using the individual word vectors together with the utterance vector. This training schema is called *Paragraph Vector Distributed memory (PV-DM)*. Figure 2 shows the *PV-DM* architecture.

Once the model is trained, it can be queried in order to get the fixed-length vector representations for each observed utterance. It can also be used to *infer* vectors for *unseen* utterances. This step is crucial to obtain the embeddings for utterances in our test dataset.

The approach presented by Le and Mikolov (2014) has two very nice advantages. Firstly, it is completely unsupervised, and thus we can use any amount of unannotated data that is available. Moreover, the model allows the use of pretrained word embeddings for initialization instead of initializing the parameters randomly.[2] This is use-

[1] Note that despite its name, the framework is able to learn embeddings for entire word sequences of any length like sentences, paragraphs or entire documents.

[2] Words that are not found in the pretrained model are still initialized randomly.

ful if there is not enough data available to learn reliable word embeddings together with the sequence embeddings. For our experiments, we try both methods: training word and paragraph embeddings entirely from scratch using different dialog corpora as input, as well as using the freely available pretrained word embeddings based on the *Google News Corpus*[3] to initialize the model.

4 Methodology

We evaluate our utterance embedding features for dialog act tagging in two different settings: 1) We tag each utterance independent of the context it appears in. 2) We include the context of the previous utterance by aggregating the utterance embeddings of the previous utterance and the current utterance to obtain a very simplified discourse model as discussed in Section 2. In addition, we experiment with two different tagset granularities. The dataset, the actual classifiers and the tagsets are described in the following two subsections.

4.1 Dialog datasets and tagsets

The dialog act classifiers are trained in a supervised fashion, which requires labeled data. For that purpose, we evaluate the actual dialog act tagging on the Switchboard Dialog Act corpus (SwDA) collected by Godfrey et al. (1992). The corpus is a compilation of telephone transcripts between two interlocutors. It contains a total of 205,000 utterances and 1.4 million words. Each utterance is assigned one of 42 possible dialog act tags according to the Discourse Annotation and Markup System of Labeling - DAMSL (Core and Allen, 1997). Table 1 shows an example of a small excerpt from the corpus with the respective dialog act labels.

In our experiments, we also use a coarser tagset where we manually map the original 42 tags into 8 classes: *Agreement*, *Answer*, *Communicative*, *Directive*, *Forward_other*, *Other*, *Statement* and *Understanding*. The mapping is described in the appendix. The motivation for this coarser tag set is we expect classifiers to more likely confuse two utterances that are close in form, for example a yes-no-question and an action-directive. If this is the case the performance of the classifiers on the coarser tag set is expected to increase substantially.

[3] https://code.google.com/p/word2vec/

We use the same training and test split as described by Stolcke et al. (2000), which uses 1,115 dialogs for training and 19 dialogs for testing and which has been widely used in previous work.[4] The utterances are preprocessed by removing linguistic annotations and any interpunctuation except for periods, question marks and exclamation marks. Additionally, as suggested by Milajevs and Purver (2014), utterances that complete previously interrupted utterances, which are marked with a continuation tag +, were concatenated to their initial segment, which also contains the correct dialog act label for the complete utterance. This is motivated by the fact that this continuation tag does not capture the ilocutionary act of the utterance, but that it shares this with the interrupted utterance.

As mentioned before, the training of the utterance embeddings is completely unsupervised and does not require any labeled data. For that reason, we experiment with expanding our data for the training of the utterance representations. In addition to the Switchboard corpus, we use the spoken dialog data from the British National Corpus (BNC) created by Clear (1993). The resulting dataset from the BNC contains approximately 1 million utterances. The BNC utterances, however, differ inherently from the Switchboard utterances as they are usually longer and normalized to full grammatical sentences. The utterances in the SWDA dataset are more fragmentary and thus usually shorter.

4.2 Classifiers

For the actual classification task we use three different classifiers from the Python *Scikit Learn*[5] package: 1) a Naive Bayes classifier, which is well suited for sparse representations like the bag-of-words vectors that we use for our baseline. 2) a K-Nearest Neighbor classifier with $K = 5$ neighbors, *euclidean distance* as distance metric and uniform weights. This technique is more suited for the dense representations like our utterance embeddings. 3) a Multilayer Perceptron with one hidden layer of 100 dimensions and a softmax layer for classification. The network was trained for 25 iterations at a learning rate of 10^{-3}.

[4] http://web.stanford.edu/~jurafsky/ws97/

[5] http://scikit-learn.org

	accuracy
Kalchbrenner and Blunsom (2013)	73.9 %
Stolcke et al. (2000)	71.0 %
Milajevs and Purver (2014)	63.9 %

Table 2: Accuracy scores from other work

5 Results

In this section we report the results for our baseline and different configurations for finding utterance embeddings. Also we present some analysis of our distributional representations.

5.1 Baseline

As a baseline we use a Bag-of-Words (BOW) representation. We represent each utterance as a sparse vector, where every dimension maps to a word in the vocabulary. We then train a Naive Bayes(NB) model and a K-Nearest Neighbor (KNN) classifier on the dialog act tagging task, both for the full tag set of 42 tags as well as on the aggregated tag set of eight tags. Note that we could not train a Multilayer Perceptron (MLP) for the baseline due to memory constraints. The results can be seen in Table 3. The strongest baseline we found was 59.98% accuracy with the NB classifier. Furthermore we calculated a trivial baseline where every utterance is classified as the most frequent dialog act tag, this would result in an accuracy of 31.47%.

To compare our results to those of the state of the art models proposed in earlier work we present their accuracy score in Table 2. These models were evaluated on the same test set.

5.2 Classification using Utterance Embeddings

We inferred embeddings for utterances as discussed in section 3. Different setting were used to find these embeddings. We varied the training data, used 50 training epochs and pretrained word embeddings. With these utterance embeddings we trained Naive Bayes (NB), K-Nearest Neighbors (KNN) and Multilayer Perceptron (MLP) classifiers. In Table 3 the results are presented. As the results suggest, none of the methods could reach the BOW baseline set up by the Naive Classifier. At least the high quality of the word embeddings is confirmed by the fact that the BOW baseline of the KNN method is outperformed by the utterance embeddings.

5.3 Using context

To use the contextual information of the dialog to classify dialog act tags, we experimented with two simple approaches. We either concatenated or summed the embedding of the previous utterance to the embedding of the current one. The first approach nicely represents the sequentiality of the dialog but it doubles the amount of features for the classifier to consider. The latter approach is not affected by this curse of dimensionality, however it does lose all sequential information present in the dialog. As expected, the latter approach would not yield any improvement. On the contrary, the tagging accuracies are worse when utterance embeddings are simply added. For that reason we did not follow that approach any further. The results for the concatenation of utterances (concat) can also be seen in Table 3.

5.4 Error Analysis

Utterance embeddings have the property of encapsulating the meaning of utterances from the words that compose them; as with the case of word embeddings, these representations present appealing relations from which similar words (in a semantic sense) appear close by in the high dimension space. We extract samples from the SWDA corpus belonging to tags that are clearly unrelated and apply dimensionality reduction to their vector representations using Principal Component Analysis (PCA). We plot the results in three dimensions. The sampled utterances belong to the following categories: *Wh-question*, *Thanking*, *Reject*, and *Apology*. Considering the words that are used in these kind of units, sample points should be clearly separated. The utterance embeddings were extracted using a 300-dimensional model trained on the SWDA corpus using pretrained word vectors. Figure 3 shows the utterance embeddings in 3D after applying PCA.

Though, the classification task seems easy considering the previous tag examples, it gets more complicated when we handle other types of utterances. Figure 4 shows the diagram after applying PCA to utterance embeddings of the five most frequent tags (*Statement-non-opinion*, *Acknowledge*, *Statement-opinion*, *Accept*, *Turn exit*), which account for 78% of the corpus.

	full tagset			aggregated tagset		
	NB	**KNN**	**MLP**	**NB**	**KNN**	**MLP**
baseline BOW	59.98%	39.57%	-	72.71%	49.61%	-
SWDA	38.18%	54.55%	50.32%	59.55%	70.82%	66.76%
SWDA+BNC	38.16%	53.86%	51.40%	60.10%	68.75%	68.39%
SWDA concat	27.24%	33.29%	51.42%	51.21%	43.87%	67.98%
SWDA+BNC concat	25.02%	34.07%	51.68%	51.68%	44.64%	68.36%

Table 3: Tagging accuracies for different models and tagsets. All models were trained with 50 iterations and pretrained word embeddings.

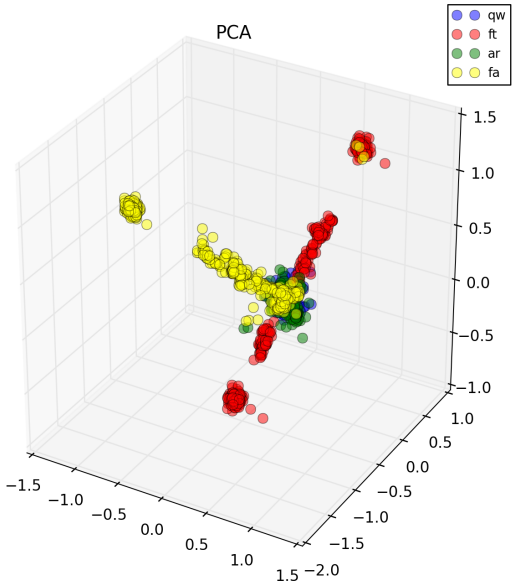

Figure 3: 3D PCA.

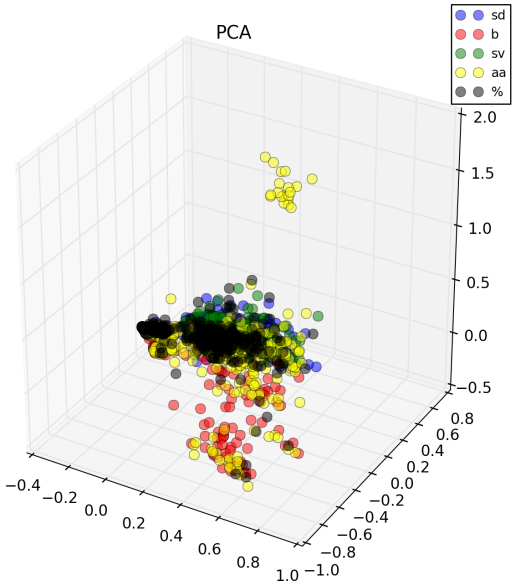

Figure 4: 3D PCA.

6 Conclusion

We see that all models trained on utterance embeddings found with the techniques from Le and Mikolov (2014) outperform the trivial baseline of predicting the most frequent tag. This supports the hypothesis that such distributional representations can be used successfully for dialog act tagging.

However a simple model using a bag of words representation and a Naive Bayes classifier outperforms these more intricate representations and models. [DW: why is this? because of the short utterances? seems strange...]

Moreover we see that training the utterance embeddings with extra samples from the BNC does not have a big effect on the accuracy of the classifiers. This could be explained by the fact that the utterances in the BNC have been cleaned to be more like grammatically correct sentences, where the Switchboard corpus is comprised of more accurately transcribed utterances. Therefore we recommend future work to use additional data that is more similar to the data used in training and evaluating the classifier in order to investigate whether this increases the accuracy of the tagging.

When comparing the results of the classifiers on a coarser tag set we see that the accuracy dramatically increases for all classifiers. This shows that the classifiers make most mistakes confusing dialog acts that are closely related. However the same goes for our baseline which has a similar increase in accuracy when predicting on a coarser tag set. Therefore we are not able to beat the baseline on a coarser tag set either.

Perhaps the most prominent deficiency of the approach presented in this paper is the lack of naturally incorporating context into the utterance representation. Using only the previous utterance is not enough to model the context of the current utterance. Even though this behaviour was to be expected, it is surprising that aggregat-

ing the previous embedding (either by concatenation or addition) did not improve accuracy significantly. Future work should therefore mainly concentrate on mending this weakness. One possible approach to this problem would be to use yet another deep learning neural network technique called Long Short Term Memory (LSTM) as proposed by Hochreiter and Schmidhuber (1997). This recurrent neural network uses a so called memory cell with four connections, an input gate, an output gate, a self recurrent gate and a forget gate. The input gate can be used to alter the state of the memory cell, while the output gate can be used to affect the state of other layers. This technique can be applied to dialog act tagging by using the utterance embeddings presented earlier as inputs for the LSTM. The output of the LSTM cell can then be used to predict a tag for that utterance. Since the utterances are fed to the LSTM in sequence, the memory cell can be thought to represent the contextual knowledge of the conversation up to that point.

From our initial hypothesis, we were able to show that: (a) Utterance embeddings capture relevant features given the words that compose them and can be used in dialog act classification; (b) The multiple experiments we run, varying training data, classifiers, and hyperparameters for all components, demonstrate that our approach is not sufficient and a simple method based on Bag-of-Words can outperform it; (c) Using additional data to get the utterance embeddings does not consistently yield better results. The final performance varies per classification technique and improvements are not of a notable magnitude.

References

John L. Austin. 1962. How to do things with words.

Jeremy H. Clear. 1993. The digital word. chapter The British National Corpus, pages 163–187. MIT Press, Cambridge, MA, USA.

Mark G Core and James Allen. 1997. Coding dialogs with the damsl annotation scheme. In *AAAI fall symposium on communicative action in humans and machines*, pages 28–35. Boston, MA.

J.J. Godfrey, E.C. Holliman, and J. McDaniel. 1992. Switchboard: telephone speech corpus for research and development. In *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, volume 1, pages 517–520 vol.1, Mar.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent convolutional neural networks for discourse compositionality. *CoRR*, abs/1306.3584.

Quoc V Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Dmitrijs Milajevs and Matthew Purver. 2014. Investigating the contribution of distributional semantic information for dialogue act classification. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*, pages 40–47, Gothenburg, Sweden, April. Association for Computational Linguistics.

Andreas Stolcke, Noah Coccaro, Rebecca Bates, Paul Taylor, Carol Van Ess-Dykema, Klaus Ries, Elizabeth Shriberg, Daniel Jurafsky, Rachel Martin, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Comput. Linguist.*, 26(3):339–373, September.

Appendix

SWBD-DAMSL	SWBD	Cnt	%	Aggregated
Statement-non-opinion	sd	72,824	36%	Statement
Acknowledge (Backchannel)	b	37,096	19%	Understanding
Statement-opinion	sv	25,197	13%	Statement
Agree/Accept	aa	10,820	5%	Agreement
Abandoned or Turn-Exit	% -	10,569	5%	Communicative
Appreciation	ba	4,633	2%	Understanding
Yes-No-Question	qy	4,624	2%	Directive
Non-verbal	x	3,548	2%	Communicative
Yes answers	ny	2,934	1%	Answer
Conventional-closing	fc	2,486	1%	Forward_other
Uninterpretable	%	2,158	1%	Communicative
Wh-Question	qw	1,911	1%	Directive
No answers	nn	1,340	1%	Answer
Response Acknowledgement	bk	1,277	1%	Understanding
Hedge	h	1,182	1%	Other
Declarative Yes-No-Question	qy^d	1,174	1%	Directive
Other	o,fo,bc,by,fw	1,074	1%	Other
Backchannel in question form	bh	1,019	1%	Understanding
Quotation	^q	934	0.50%	Other
Summarize/reformulate	bf	919	0.50%	Understanding
Affirmative non-yes answers	na,ny^e	836	0.40%	Answer
Action-directive	ad	719	0.40%	Directive
Collaborative Completion	^2	699	0.40%	Other
Repeat-phrase	b^m	660	0.30%	Understanding
Open-Question	qo	632	0.30%	Directive
Rhetorical-Questions	qh	557	0.20%	Directive
Hold before answer/agreement	^h	540	0.30%	Agreement
Reject	ar	338	0.20%	Agreement
Negative non-no answers	ng,nn^e	292	0.10%	Answer
Signal-non-understanding	br	288	0.10%	Understanding
Other answers	no	279	0.10%	Answer
Conventional-opening	fp	220	0.10%	Forward_other
Or-Clause	qrr	207	0.10%	Directive
Dispreferred answers	arp,nd	205	0.10%	Answer
3rd-party-talk	t3	115	0.10%	Communicative
Offers, Options Commits	oo,cc,co	109	0.10%	Other
Self-talk	t1	102	0.10%	Communicative
Downplayer	bd	100	0.10%	Understanding
Maybe/Accept-part	aap/am	98	<.1%	Agreement
Tag-Question	^g	93	<.1%	Directive
Declarative Wh-Question	qw^d	80	<.1%	Directive
Apology	fa	76	<.1%	Forward_other
Thanking	ft	67	<.1%	Forward_other

SWDA corpus aggregated tag set