



Dialogue Act Tagging and Segmentation with a Single Perceptron

Ramon Granell¹, Stephen Pulman¹
 Carlos-D. Martínez-Hinarejos², José Miguel Benedí²

¹Computing Laboratory, University of Oxford, Oxford, United Kingdom

²Instituto Tecnológico de Informática, Universidad Politécnica de Valencia, Valencia, Spain

ramon.granell@comlab.ox.ac.uk, sgp@clg.ox.ac.uk
 cmartine@dsic.upv.es, jbenedi@dsic.upv.es

Abstract

In this paper we present a simultaneous automatic Dialogue Act (DA) tagger and segmenter. The model employed is based on the well-known single layer perceptron algorithm used successfully in other Computational Linguistic tasks. A decoding process was developed for searching the sequence of segments and DA tags from all the possible exponential possibilities. A set of features based on combination of words and DA tags were empirically selected. Models were tested over transcriptions of two corpora of dialogues (Switchboard and Dihana) and transcriptions and ASR output of a third corpus composed by meetings (AMI corpus). The results obtained for such a simple but powerful model are for some of the evaluation metrics equal or better than much more complex models presented in recent studies for the same experiments.

Index Terms: dialogue act segmentation, dialogue act tagging, single perceptron

1. Introduction

Detecting Dialogue Acts (DA) from speaker turns is an essential task for many Dialogue Systems and useful to solve other Natural Language Processing (NLP) problems such as Automatic Speech Recognition (ASR) [9], Speech Synthesiser [14] or automatic summarization of conversations [8]. DAs are based on the concept of speech acts that originally comes from linguistic pragmatic theory and they basically define the speaker intention involved in the utterance. Several DA annotation schemes exist which define their own DA tag set. However, most of these DA schemes do not limit the DA tag to just one turn, being possible to have several DAs in the same dialogue turn. Therefore, we need to simultaneously segment a raw utterance and assign a DA tag to each segment.

In recent years, some stochastic machine learning approaches have been employed for solving both the DA tagging and segmentation simultaneously. In [5] they use a model based on a switching dynamic Bayesian network (DBN) architecture and evaluate it over multiparty meetings. [7] shows results employing two different models: the first one based on a combination of Hidden Markov Models and N-grams and the second model based on Transducers with N-grams as well.

In this work, we study how to solve these problems simultaneously using a discriminative model based on the single perceptron algorithm and we compare the results obtained over three corpora with recent previous work. The perceptron algorithm is a well-known linear model that recently has been used in many NLP tasks such as Named Entity Recognition [4], and Part-Of-Speech labelling [12] obtaining competitive results

with respect to more complicated methods. Additionally, a multilayer perceptron model has been previously used to solve the problem of DA tagging [13] but not both tasks simultaneously.

2. Model for DA tagging and segmentation

We can define the problem of combined DA tagging and segmentation in the following way. Let $t = \{W, U, D\}$ a DA tagged dialogue turn composed of a sequence of words $W = w_1 \dots w_l$ that are grouped into n segments (utterances) according to the indexes $U = u_0 u_1 \dots u_n$, where any $u_i \in \mathbb{N}, 0 \leq i \leq n, 0 \leq u_i \leq l, u_0 = 0, u_n = l$ and $u_{j-1} < u_j$ with $1 \leq j \leq n$. The i -th segment will be formed by the words $w_{u_{i-1}+1} w_{u_{i-1}+2} \dots w_{u_i-1} w_{u_i}, 1 \leq i \leq n$. Therefore, W can be rewritten as

$$W = w_{u_0+1} \dots w_{u_1-1} w_{u_1} \dots w_{u_{n-1}+1} \dots w_{u_n-1} w_{u_n}.$$

Finally, D is the sequence of Dialogue Act tags $D = d_1 \dots d_n$ where each DA tag $d_i \in \Delta$ corresponds to the DA of the i -th segment, and Δ is the set of different DA tags. Then, given as an input this sequence of words W , we will look for the sequence of segments and DA tags, t^* that satisfies:

$$t^* = \operatorname{argmax}_{t' \in \text{GEN}(W)} \text{Score}(t') \quad (1)$$

where $\text{GEN}(W)$ is the set of all possible combinations of segments and DA tags obtained from the sequence of words W and tag set Δ , $\text{GEN}(W) = \{t' = \{W, U', D'\}, \forall \{U', D'\}, U' = u'_0 u'_1 \dots u'_{n'} \wedge u'_0 = 0, u'_{n'} = l, D' = d'_1 \dots d'_{n'}, d'_i \in \Delta \wedge 1 \leq n' \leq l\}$. $\text{GEN}(W)$ is called the set of candidates as well. $\text{Score}(t')$ is a linear function:

$$\text{Score}(t') = \vec{v} \cdot \phi(t') \quad (2)$$

where $\phi(t') \in \mathbb{N}^k$ is the vector that represents the features of the candidate t' , $\vec{v} \in \mathbb{R}^k$ is a vector of weights (they are the parameters of the model) and \cdot is the vector inner (dot) product. Using the perceptron algorithm it is possible to learn the vector \vec{v} (see Figure 1). We make use of the averaged perceptron algorithm, [3], an extension of the perceptron algorithm which better avoids the problem of overfitting.

2.1. Features

The features vector $\phi(t) \in \mathbb{N}^k, \phi = \phi_1, \dots, \phi_k$ obtained from each segmented and tagged turn or candidate $t = \{W, U, D\}$ is a set of global contextual features. In our model, we define a **local feature** f_i for a segment and its DA tag $\{w_{u_{j-1}+1} \dots w_{u_j}, d_j\} \in t$ as a function that can be of the following two different types:

Input: set of tagged and segmented dialogue turns T ,
where each $t \in T$, $t = \{W, U, D\}$,
set of different DAs Δ
Output: vector of weights $\vec{v} \in \mathbb{N}^k$
Initialization: $v_i = 0$, $1 \leq i \leq k$

- 1: **for** $i = 1 \dots I$ **do** // I is the number of iterations
- 2: **for all** $t \in T$ **do**
- 3: $t^* = \operatorname{argmax}_{t' \in \text{GEN}(W)} \vec{v} \cdot \phi(t')$
- 4: **if** $t^* \neq t$ **then**
- 5: $\vec{v} = \vec{v} + \phi(t) - \phi(t^*)$
- 6: **end if**
- 7: **end for**
- 8: **end for**

Figure 1: Perceptron learning algorithm.

1) A binary function indicating the presence or absence of a set of conditions over the words and tags, e.g.

$$f_3(\{w_{u_{j-1}+1} \dots w_{u_j}, d_j\}) = \begin{cases} 1 & \text{if } \exists i, u_{j-1} + 2 \leq i \leq u_j / w_{i-1} = 'I' \wedge w_i = 'see' \\ & \wedge d_j = 'Appreciation' \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

2) A summation of a binary function over all the words that formed the segment, e.g.

$$f_{271}(\{w_{u_{j-1}+1} \dots w_{u_j}, d_j\}) = \sum_{i=u_{j-1}+2}^{u_j} f_3(\{w_{i-1}, w_i, d_j\}) \quad (4)$$

Some of these features can also use information from the context such as words of previous segments ($w_{u_{j-2}+1} \dots w_{u_{j-1}}$) or previous DA tags (d_{j-2}, d_{j-1}).

Global contextual features are defined over the whole turn t and simply are the sum over the local features of all the segments of this turn:

$$\phi_i(t) = \sum_{\{w_{u_{j-1}+1} \dots w_{u_j}, d_j\} \in t} f_i(\{w_{u_{j-1}+1} \dots w_{u_j}, d_j\}) \quad (5)$$

In this work, we have selected a set of features based on previous studies [10, 12], refining them through experimentation. Finally, 28 feature templates based on the lexicon, segmentation and DA tagging were selected. Most of them can be seen in Table 1.

2.2. Decoding process

The decoding algorithm is used both to estimate the best candidate for updating the weights in the perceptron learning algorithm and to find the best combination of DA tags and segments during the test. In the set $\text{GEN}(W)$ there are all the possible combinations of DA tags and segments given the words $W = w_1 \dots w_{|W|}$. That means that for the $|W|$ words that form the whole turn there are $2^{|W|-1} |\Delta|^{|W|}$ combinations of segments and DAs, where Δ is the set of different DA tags. As this exponential explosion is not computationally tractable, a beam-search algorithm was implemented. A dynamic programming algorithm can not be implemented because the nature of some of the features of the candidates such as the ones regarding to previous segments and DA tags to the last one.

Segment formed by words $w_1 \dots w_{ W }$
Two consecutive segments formed by words $w_1 \dots w_{ W }$ and $w'_1 \dots w'_{ W' }$ respectively
Segment formed with $ W $ words starting/ending with word w
Segment with last word w followed with segment starting with word w'
Number of times word bigram ww' appears in the segment
Segment with first word w and last word w'
Two consecutive segments starting/ending respectively with word w and w'
Segment formed with $ W $ words with previous segment formed with words $w_1 \dots w_{ W' }$
Segment formed with $ W $ words with next segment formed with words $w_1 \dots w_{ W' }$
Segment formed by specific words $w_1 \dots w_{ W }$ with DA tag d
Two consecutive DA tags dd'
Three consecutive DA tags $dd'd''$
Two consecutive segments, the first one formed with specific words $w_1 \dots w_{ W }$ and the second with DA tag t
Segment formed with words $w_1 \dots w_{ W }$ and tag d immediately after/before word w'
Segment formed with first/last word w and DA tag t
Number of times word w appears in the segment in neither the first nor last position and its DA tag is t
Number of times word w appears in the segment in not the first position and its DA tag is t and first segment word is w'

Table 1: Some of the feature templates.

The decoding algorithm [12] basically consists of going incrementally from the first word of the turn until the last one ($w_1 \dots w_{|W|}$), storing the n best candidates (the ones with the highest value for Score function) for each of the subsequence of words $w_1 \dots w_i$, $1 \leq i \leq |W|$ in a storage structure called the *agenda*. Therefore, $agenda[w_i]$ will be formed by the n best candidates for the sequence of words $w_1 \dots w_i$. These are obtained by combining a segment ending in w_i with the candidates stored in previous agendas ($agenda[w_j]$, $1 \leq j < i$). Final results for the whole turn will be in $agenda[w_{|W|}]$.

3. Conversation data

The model explained in the previous section has been tested over three corpora. The Dihana corpus [1] is a set of Human-System (Wizard of Oz) dialogues in Spanish about railway information for timetables and trains. Five partitions of 180 dialogues each one were defined. The second corpus is the Switchboard corpus (SWBD) [6], that is formed by 1155 Human-Human English dialogues by phone about freely chosen topics. 11 partitions each one of 105 dialogues were defined. Partitions over these corpora are the same as the ones defined in [7]. The last of the corpora is the AMI Meeting corpus [2] that is a set of meetings each one with four participants that discuss development of a product. Three partitions were defined in the corpus documentation: 98 meetings in the training set, 20 in the development set and 20 in the test set. These partitions were the same as used in [5].

All the conversations of these corpora were manually transcribed, and later their turns segmented and DA annotated, with each corpus using a different DA scheme. The Dihana corpus was annotated using tags with three levels of granularity, using in this work the first and second (Dihana.2lev) and all three lev-

Corpus / Features	Voc. size	#Diff. DAs tags	#Dialogues	#Turns	avg. segm/turn	#Partitions
Dihana_2lev	~900	72	900	15,413	1.3	5
Dihana_3lev	~900	248	900	15,413	1.5	5
SWBD	~42,000	42	1,155	115,255	1.8	11
AMI	~10,400	15	138	66,354	1.6	3
AMI-ASR	~9,200	15	138	56,923	1.8	3

Table 2: Some features of the corpora and partitions selected to perform experiments.

Metric	Dihana_2lev			Dihana_3lev			SWBD			
	HMM-NG	NGT	PER	HMM-NG	NGT	PER	HMM-NG	NGT	PER	
Segm.	NIST-SU	43.8	1.2	2.4	45.7	4.2	4.4	45.2	24.0	21.2
	DSER	44.1	1.9	2.6	46.3	6.9	6.0	61.8	35.9	27.0
	SegER	22.9	1.1	2.3	25.6	4.0	4.0	41.2	22.9	18.4
	Strict	52.3	1.2	2.4	57.9	4.9	4.6	71.8	47.0	40.1
	Boundary	4.5	0.1	0.2	5.3	0.5	0.5	4.9	2.6	2.3
Tag.	DAER	7.9	7.9	6.5	15.3	17.4	22.4	54.6	46.6	36.1
	Lenient	9.0	8.7	5.0	13.3	18.2	13.6	30.9	38.8	27.6
Segm. and tag.	NIST-SU	50.2	8.3	7.1	53.8	18.2	23.3	64.2	52.2	43.8
	DER	49.2	8.5	6.3	51.1	18.7	22.5	71.4	57.7	44.4
	SegDAER	29.2	8.3	6.9	33.6	17.9	22.8	60.4	50.4	40.8
	Strict	57.4	9.1	6.1	61.4	19.2	14.7	79.8	66.8	54.8

Table 3: Results of experiments with the Dihana and SWBD corpora. HMM-NG is the combination of Hidden Markov Models and N-grams and NGT is the Transducer with the N-gram [7]. PER is the model of the perceptron algorithm presented in this work.

els (Dihana_3lev). Different annotation schemes using 42 and 15 DA tags were employed for tagging the transcriptions of the SWBD corpus and the AMI corpus, respectively. Additionally, Automatic Speech Recognition output for the meetings of the AMI corpus is also available (AMI-ASR) with a Word Error Rate over 26%. To obtain the reference DA tags and segments for AMI-ASR an alignment between the time stamps of each conversation and the DA tags that correspond to the manual transcriptions was performed. This alignment has the problem that many words of the ASR are temporally between two DA segments of the transcriptions (they start in one segment and end in another one). In order to solve this problem automatically, these words are compared with the real transcriptions to decide to which segment corresponds. More details of all the corpora and their tagging can be seen in Table 2.

4. Experiments

4.1. Development experiments

Development experiments tested over small sets of data (the development set for the AMI corpus and the first of the partitions for the other corpora) were performed for selecting the features, the *agenda* size and the final number of iterations for the perceptron algorithm. Finally, for the Dihana corpus an *agenda* of size 16 was employed in the decoding process and for the SWBD and AMI corpora the *agenda* size was 4. The algorithm converges within a small number of iterations (less than five) for all corpora.

4.2. Evaluation metrics

Several evaluation metrics have been used for analysing the performance of both DA tagging and segmentation. In [7] they make use of edit distance over DA tags (**DAER**), segment boundaries (**SegER**) and combining both of them (**SegDAER**).

However, in other works [5, 13] they evaluate their models with other metrics:

- Segmentation metrics. **NIST-SU** is the number of errors in the segment boundary calculated word by word compared with the reference segmentation and normalized by the total number of words. **Boundary** is the same boundary errors that NIST-SU but normalized by the number of segments of the reference. For **DSER**, the number of error segments is calculated (an error segment is when the segment does not start and end exactly in the same word than the reference segmentation) and normalized by the number of segments in the reference. **Strict** metric is the number of words that are in error segments divided by the number of words.
- Tagging metric. **Lenient** is the number of words that are not assigned the correct DA tags, regardless of the segmentation and divided by the total number of words.
- Segmentation and tagging metrics. **NIST-SU**, **DER** and **Strict** are respectively calculated in the the same way that NIST-SU, DSER and Strict for segmentation but taking also into account the DA tag of the word and segment to calculate the errors.

We calculate many different metrics in order to facilitate comparison with other reported works [5, 7].

4.3. Results

Cross-validation experiments were performed for both the Dihana and the SWBD corpora as in [7] using partitions defined in Table 2. Comparative results using all the metrics previously defined can be seen in Table 3. Scores for the Dihana_2l and Dihana_3l corpora are quite similar for both approaches. However, SWBD results obtained with the model presented here are

Metric	AMI					AMI-ASR					
	iFLM	Hyb	iFLM_np	Hyb_np	PER	iFLM	Hyb	iFLM_np	Hyb_np	PER	
Segm.	NIST-SU	20.4	25.6	31.9	51.8	26.1	30.7	34.0	45.6	70.9	49.2
	DSER	12.8	17.0	24.5	36.0	35.1	23.2	25.8	47.8	62.1	57.6
	Strict	28.5	36.9	50.7	63.2	42.2	26.9	33.7	51.2	67.5	70.6
	Boundary	3.1	3.9	4.9	7.9	3.3	5.0	5.5	7.4	11.5	6.7
Tag.	Lenient	51.8	42.2	51.4	44.0	37.9	57.1	46.9	55.9	49.7	47.0
Segm. and tag.	NIST-SU	73.6	71.3	85.4	102.2	58.5	84.0	81.2	99.2	123.4	84.6
	DER	57.0	51.9	61.8	61.7	58.7	68.6	64.1	85.3	87.1	80.1
	Strict	64.4	62.1	74.8	77.1	62.0	68.3	64.7	78.4	82.3	85.3

Table 4: Results of experiments with the AMI corpus. iFLM and Hyb corresponds respectively to the Interpolated Factored Language Model (FLM) and a hybrid model of iFLM and FLM presented in [5]. iFLM_np and Hyb_np are the same than iFLM and Hyb models but not using prosodic features. PER is the model of the perceptron algorithm explained in this work.

significantly better than in [7]. This may be due to the large difference in vocabulary size in the two corpora.

In the case of the AMI corpus experiments were performed using the partitions defined in the previous section and as in [5]. Results for these experiments are shown in Table 4. For the transcriptions and for some of the segmentation and tagging metrics such as NIST-SU, Strict and Lenient results are better than the models in [5] in spite of the fact that they make use of prosodic features. This is due to the low number of false negatives that it produces. However, results for AMI-ASR are in general worse as our model is strongly based on the vocabulary. Additionally, this AMI-ASR corpus might be slightly different than the one in [5] because of the alignment process described in Section 3.

5. Conclusions and future work

In this work we have presented a model for DA tagging and segmentation using a single layer perceptron. We have evaluated the model using three different corpora, obtaining competitive results compared to some of the most recent work in the field. Obviously, as was shown in [5] an important improvement in the accuracy in the task can be achieved using prosodic features, which seem easy to incorporate in the present model as a new feature. Other features that can be used for the transcriptions and were previously used in other studies [10] are syntactic features.

An immediate objective of this work is using this technique in a real Dialogue System prototype. In the Companions project [11] one of the modules of the whole Dialogue System called Dialogue Act Tagger and Segmenter (DAT) performs exactly the task described in this paper. At the moment, error rates are very high for real ASR input. However, in a Dialogue System we know exactly the words and DA tags of the system turn, and this can be used to improve the performance of the DA tagging and segmentation for user turns as it happens with the Dihana corpus. For using in a real speech application the influence of the accuracy of the DAT should be evaluated over the modules that receive its output such as the Natural Language Understanding module and finally over the whole integrated system.

6. Acknowledgments

This work was partially funded by the Companions project (www.companions-project.org) sponsored by the European Commission as part of the Information Society Technologies (IST) programme under EC grant number IST-FP6-034434.

7. References

- [1] Benedí, J.M., Lleida, E., Varona, A., Castro, M.J., Galiano, I., Justo, R., López de Letona, I., Miguel, A. 2006. Design and acquisition of a telephone spontaneous speech dialogue corpus in Spanish: Dihana. In: Fifth International on LREC
- [2] Carletta, J.C., Ashby, S., Bourban, S., Flynn, M., Guillemot, M., Hain, T., Kadlec, J., Karaiskos, V., Kraaij, W., Kronenthal, M., Lathoud, G., Lincoln, M., Lisowska, A., McCowan, M., Post, W., Reidsma, D., and Wellner, P. 2005 The AMI Meeting Corpus: A Pre-Announcement. 2nd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms, Edinburgh, UK
- [3] Collins, M. 2002. Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithms. In Proc. of the ACL-02 Philadelphia, USA
- [4] Collins, M. 2002. Ranking algorithms for named-entity extraction: boosting and the voted perceptron. In Proc. of the ACL-02 Philadelphia, USA
- [5] Dielmann, A. Renals, S. Recognition of Dialogue Acts in Multiparty Meetings Using a Switching DBN. 2008. Audio, Speech, and Language Processing, IEEE Transactions on, Vol. 16, No. 7.
- [6] Godfrey, J., Holliman E., McDaniel J. 1992. SWITCHBOARD: telephone speech corpus for research and development. ICASSP
- [7] Martínez-Hinarejos, C.D. Tamarit, V. Benedí, J.M. 2009 Improving unsegmented dialogue turns annotation with n-gram transducers. In Proc. of the 23rd PACLIC23. Hong Kong, China.
- [8] Murray, G. and Carenini, G. 2008. Summarizing spoken and written conversations. In Proc. of the ACL 2008 Conference. Honolulu, USA
- [9] Stolcke, A., Coccaro, N., Bates, R., Taylor, P., van Ess-Dykema, C., Ries, K., Shriberg, E., Jurafsky, D., Martin, R., Meteer, M. 2000. Dialogue act modelling for automatic tagging and recognition of conversational speech. Computational Linguistics 26 (3)
- [10] Verbree, A.T. and Rienks, R.J. and Heylen, D.K.J. 2006 Dialogue-act tagging using smart feature selection: results on multiple corpora. In: First International IEEE Workshop on SLT 2006, Palm Beach, Aruba.
- [11] Wilks, Y. 2006. Companions: Intelligent, persistent, personalised interfaces to the internet. <http://www.companions-project.org>
- [12] Zhang Y., Clarck S. 2008. Joint Word Segmentation and POS Tagging Using a Single Perceptron. Proceedings of ACL-08: HLT. Columbus, USA
- [13] Zimmermann, M., Hakkani-Tür, D., Shriberg E., Stolcke A. 2006 Text Based Dialog Act Classification for Multiparty Meetings. In Proc. of MLMI
- [14] Zovato E. and Romportl J. 2008. Speech synthesis and emotions: a compromise between flexibility and believability. Fourth International Workshop on Human-Computer Conversation Bellagio, Italy