A SVM Cascade for Agreement/Disagreement Classification

Pierre Andrews* — **Suresh Manandhar****

* *andrews@disi.unitn.it;* ** *suresh@cs.york.ac.uk*
* *Dipartimento di Ingegneria e Scienza dell'Informazione*
Università degli Studi di Trento
38050 Trento, Italy
** *Department of Computer Science*
The University of York
YO105DD York, United Kingdom

ABSTRACT. *This article describes a method for classifying dialogue utterances and detecting the interlocutor's agreement or disagreement. This labelling can help improve dialogue management by providing additional information on the utterance's content without deep parsing. The proposed technique improves upon state of the art approaches by using a* Support Vector Machine cascade. *A combination of three binary support vector machines in a cascade is employed to filter out utterances that are easy to classify, thus reducing the noise in the learning of labels for more ambiguous utterances. The approach achieves higher accuracy (by 2.47%) than the state of the art while using a simpler approach which relies only on shallow local features of the utterances.*

RÉSUMÉ. *Dans cet article, nous décrivons une méthode de classification d'uttérances destinée à la detection d'accord/désaccord dans le dialogue homme-machine. L'étiquetage du dialogue peut être utilisé par le dialogue manager sans avoir à effectuer de parse complexe. Nous proposons une technique de classification à base d'une hiérarchie de classificateurs Support Vector Machines. Une combinaison de trois classificateurs binaires est utilisée pour filtrer les classes pour lesquelles le corpus contient beaucoup d'information et se concentrer sur les classes plus ambigües. Cet article présente une analyse détaillée des traits caractéristiques de classification et propose une amélioration de 2.47% sur l'état de l'art tout en utilisant un modèle de classification plus performant.*

KEYWORDS: *SVM, Dialogue, Agreement, Disagreement, Opinion, Classification.*

MOTS-CLÉS : *SVM, dialogue, accord, désaccord, opinion, classification.*

TAL. Volume 50 – n° 3/2009, pages 89 à 107

1. Introduction

Human-Computer Dialogue is a major field of research in natural language processing. In addition to understanding the language inputs of the user, it is also very important to manage the flow of the dialogue to generate a conversation as natural as possible. However, the strict rules of interaction in human conversations can help in developing strategies to simplify the natural language understanding required to follow a conversation.

In this article, we discuss the application of a supervised learning algorithm applied to argumentative dialogue management. In this type of dialogue, even if a full understanding of the user utterances cannot be achieved, being able to detect agreement vs. disagreement utterances can substantially aid the robustness of the dialogue system. We thus developed a classifier model that can label the users' utterances within four classes: agreement, disagreement, other or backchannel. [1] This labelling can then be used to manage the dialogue according to the user's reactions.

We introduce a classification model based on shallow features of utterances combined with a support vector machine classifier. The proposed model improves (by 2.47%) on existing state of the art approaches and achieves 86.5% accuracy when classifying the dialogue utterances.

In this article we explore the state of the art in classification of dialogue utterances as *Agreements* or *Disagreements*. After a study of the possible features that can be used to characterise each class, we expand on the work from Hillard *et al.* (2003) and Galley *et al.* (2004) by proposing to simplify the feature set and to perform supervised learning with a Support Vector Machine (SVM). The classification described in this article is aimed at helping the management of argumentative human-computer dialogues.

This article is structured as follows. Section 2 first introduces the application of this classifier to the dialogue management task. Section 3 explores the existing classification methods developed in the state of the art for this task and their conclusions. Section 4 extends this state of the art with a statistical analysis of the linguistic features that can be used to characterise the *Agreement* and *Disagreement* classes. The statistical study is performed on a manually annotated corpus of 8135 dialogue utterances that extends the ICSI Meetings Corpus (Janin *et al.*, 2003) provided by Galley *et al.* (2004).

Section 5 describes the features that are used by our learning algorithm and the cascade of binary SVM classifiers that we use for classification in the dialogue task. Section 6 shows that by using a simplified set of features to train a more sophisticated classifier we can improve on the state of the art approaches and obtain an accuracy of 86.53% on Galley *et al.*'s (2004) model. This section also provides a detailed analysis

1. Dialogue utterances that do not carry pragmatic content in the dialogue (see next section).

of the results and discussion of possible improvements to the approach to increase the classification accuracy.

This article shows a possible application of automatic machine learning to dialogue management and demonstrates how a classifier can be built by doing a strong analysis of the available features, based on linguistic theories as well as an empirical analysis of a manually annotated corpus. We show that by using a feature set grounded in a sound analysis of the utterance characteristics combined with a state of the art learning algorithm it is possible to train a classifier for detecting *Agreement* and *Disagreement* utterances in a dialogue.

2. Classification for Dialogue Management

In the field of human-computer dialogue, the process of understanding the user, keeping a model of the user's beliefs and deciding what dialogue move to take next is managed by a so-called Dialogue Manager.

In this article, we concentrate on the aspect of the dialogue manager that interprets the user's utterances to help it decide its own reactions. For example, when dealing with natural argumentative dialogue (e.g. Mazzotta *et al.*, 2007; Cassell and Bickmore, 2002), the dialogue management system creates a dialogue plan, setting the arguments it wants to present to the user. However, after presenting the arguments to the user, the system needs to interpret the answer and decide if the user disagrees with the system or accepts its conclusions.

Gilbert *et al.* (2003) propose to deal with natural argumentation dialogue by implementing a deep understanding of the user's utterances. Their proposed dialogue system needs to understand the structure of the argument, the facts presented and their veracity. This requires extensive computation and remains theoretical, as such algorithms have yet to be developed.

In fact, in current dialogue systems, shallow understanding of the user utterances is preferred. By limiting the domain of the dialogue and formulating the system utterance in specific ways, the reactions of the users are limited by natural discourse rules. Levin and Moore (1977) formalise this idea with *dialogue games*, where the dialogue is described as a game with a limited set of moves. If the user chooses utterances outside of these moves, the latter can be considered errors, as they go against accepted human discourse rules.

For instance, in the field of natural argumentative dialogue, in Andrews *et al.* (2008), we proposed to simplify dialogue management by considering the limited set of moves available during argumentation. The system we introduced uses a dialogue game where argumentative dialogue moves can be classified under three categories (see figure 1):

Agreeing utterances, where the user accepts the system's argument;

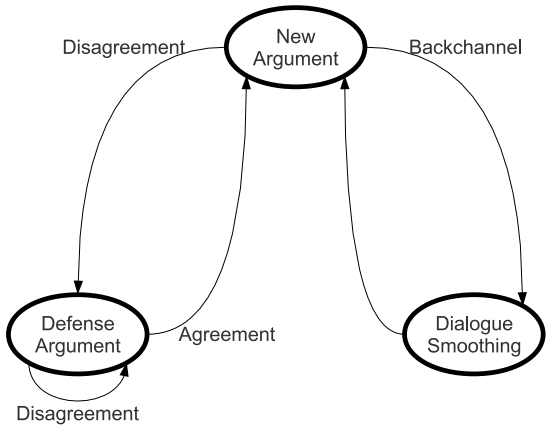

Figure 1. *Dialogue Game Transitions based on the detection of Agreement and Disagreement. The dialogue management can be divided into three dialogue phases: a)* New Argument*, where the dialogue manager initiates a new dialogue phase, discussing a new argument, b)* Defend Argument*, where the dialogue manager tries to support the current argument against the user's doubts, and c)* Dialogue Smoothing*, where the dialogue manager introduces dialogue cues to keep the user motivated by the dialogue*

Disagreeing utterances, that provide additional argumentation from the user and require that the system elaborates on its current argument's conclusion;

Back-Channel utterances, dialogue fillers that do not provide pragmatic content in the dialogue but help smooth the conversation and make it feel more natural.

In this dialogue game, if the users make an *Agreement* move, the system assumes that they agree with its argument and the system can thus shift to another argument. However, when the user makes a *Disagreement* move, the system tries to defend the current argument.

The ability to segment an interaction into these categories of utterances relies on being able to detect whether the user is agreeing with the system or rejecting the argument. Although this can be managed by domain-specific pattern matching, an automatic, domain-independent classification would allow for a more portable dialogue management system.

3. Related Work

The field of utterance classification for dialogue management is not new, and previous research has proposed different types of labelling tailored to particular application domains. Machine learning for dialogue management is often used for high-level dialogue act classification in domain-specific dialogues; in particular, the research

focuses on using prosodic cues for helping spoken dialogue interpretation (for example Stolcke *et al.* (1998), Stolcke *et al.* (2000), Fernandez and Picard (2002)). Automatic tagging of utterances can help decide where in the dialogue the system is and what next dialogue move to take (for example Andernach (1996)) but it can also be used to provide structural information to understand the content of the utterance (for example, Dinarelli *et al.* (2009)). The classification approaches are usually based on machine learning techniques for classification in known classes, but some research (for example Andernach (1996)) also tries to use machine learning techniques to identify inherent classes from the utterance features.

In this section, we focus on the existing specific research for classifying Agreement and Disagreement utterances in human dialogues. This can be compared to the new field of opinion detection, but only focuses on very shallow (i.e. binary) representation of the user's opinion.

Hillard *et al.* (2003) proposed a first step towards a statistical method for agreement/disagreement classification by developing a supervised learning classifier based on an annotated selection of meetings from the ICSI Meeting corpus (Janin *et al.*, 2003). The ICSI corpus is a collection of transcripts of meetings, which contains prosodic annotation in addition to the content of the dialogues. Hillard *et al.* selected 1800 segments of transcribed speech, called *spurts*, that correspond to segments of the dialogue with no pauses in the speech. These spurts were manually labelled with one of four possible labels:

BackChannel are short spurts that, having the form of agreement – e.g. "yeah", "ok", "yep" – could also be "encouragement for the speaker";

Positive is used for spurts that are clear agreements;

Negative is used for disagreement spurts;

Others are long spurts that cannot be classified as either agreement or disagreement.

The classifier proposed by Hillard *et al.* uses a *decision tree* algorithm with a combination of spurt features. These are:

– *lexical* features:

- the number of words;
- the number of positive/negative words;
- the Agreement/Disagreement class of the first word of the sentence.

The class of the words was inferred from their frequency in each class of labelled spurts.

– *prosodic* features:

- duration of pauses in the spurt;
- duration of the spurt;
- fundamental frequency ($F0$).

Galley *et al.* (2004) extend Hillard *et al.*'s (2003) approach by adding a number of novel features and a spurt classifier based on a Bayesian network. While Hillard *et al.* use only *local* features of a spurt to decide its class, in Galley *et al.* the feature set is extended with features from previous spurts in the dialogue to infer the class of the current spurts. We call these "global features" in the rest of the article. Galley *et al.* use *adjacency pairs* to encode the interaction between speakers and the relationship between consecutive spurts. Instead of only using features of the currently considered spurt, Galley *et al.* also use features from the general dialogue context to take into account the discourse structure when classifying.

Galley *et al.* use an *adjacency pair* feature to label each spurt with the previous spurt it relates to. For example, if one interlocutor asks a question (Q) and another interlocutor answers (A) this question directly, there is an adjacency pair $Q \rightarrow A$. This provides extended information on where the spurt is used and gives more clues to the classifier about the class of the spurt. By using sequential analysis of adjacency pairs, Galley *et al.* add *global* features, where the labelling of a spurt depends on the *agreement/disagreement* label of the previous spurt in the adjacency pair.

Related spurts may not be directly adjacent in the dialogue and other utterances may be interleaved; for example, in the previous question/answer example, the question Q might not be directly adjacent to its answer A: $Q \ldots B \ldots A$. To detect the *adjacency pairs* relationship between spurts, Galley *et al.* (2004) use a statistical ranking algorithm based on maximum entropy. Given the latest spurt of a pair (A in our example), the algorithm can learn, with 90.2% accuracy, how to find the previous element of the pair (Q in our example) in the dialogue.

The detected adjacency pairs are combined with local features in a *Bayesian network* classifier that labels spurts as either *agreements* or *disagreements*. The classifier is then trained/tested on an extended annotated corpus of 8135 spurts using the *contextual* features combined with a set of *durational* features (length of the spurt, length of silences in the spurt, etc.) and with *lexical* features (number of words, positive polarity adjectives, etc.).

Galley *et al.* (2004) show that this model can improve on Hillard *et al.*'s (2003) classification results with similar *local* features by simply using a better classification algorithm. By adding *global* features, they show an improvement of only 1% accuracy while relying on an *adjacency pair* detection algorithm with a 9.8% error rate.

In this article, we explore improvements to the accuracy of the classifier that are linked to simple *local* features, which do not require complex computation.

4. Discriminating Features

To our knowledge, there is no specific research in linguistics on the structure of agreement and disagreement utterances that could support the *local* features selected by the previous work in the state of the art. However, the adjacency pair organisation

of argumentative dialogue used by Galley *et al.* (2004) for their "global" features was discussed by Jackson and Jacobs (1980).

We are interested in developing a classifier that can rely only on "local" features. In addition, Hillard *et al.* (2003) show that prosodic features are not useful for improving the classification of agreement/disagreement utterances. Hence, in this article, we present a classifier using local features of the spurts for classification, and show that the proposed approach achieves results comparable with the state of the art without having to rely on complex features extraction such as *adjacency pairs* identification (see Section 6). Such complex features might need computation time that is not compatible with online dialogue with a user. In addition, as Galley *et al.* (2004) show, they might yield more errors in the extraction process.

The local features used in our model are equivalent to the ones used in Hillard *et al.* (2003) and Galley *et al.* (2004). In this section, we provide an empirical grounding of these features.

4.1. *Length of Utterance*

We did not find any theoretical evidence in the literature that the different classes of utterances used would be of significantly different length. However, there is strong statistical evidence in the annotated corpus that the utterance length is significant for characterising the *BackChannel* and *Agreement* classes.

Indeed, figure 2 shows the different probability densities for a spurt having a specific length for each class. We observe that the *BackChannel* spurts are significantly shorter ($M = 14$, $SD = 12$, $\hat{d} = 0.77$)[2] than all the other classes ($t(1416) = -28.3$, $p < 0.001$).[3] The *Agreement* class is also significantly shorter ($M = 108.99$, $SD = 190.54$, $\hat{d} = 0.33$) than the *Disagreement* and *Other* classes ($t(223) = -4.3$, $p < 0.001$).

The *Disagreement* and *Other* spurts are longer spurts where the interlocutors say more and provide more support for their arguments; whether they are neutral discussions or disagreements, they are not significantly different in length ($t(134) = 1.46$, $p = 0.147$).

The *length of spurt* feature can thus be an interesting feature for discriminating the *BackChannel* and *Agreement* classes when classifying.

2. M is the mean length, SD is the deviation in length and $\hat{d}$ is the effect size computed with Cohen's d.

3. t is the value of the independent two-sample *t*-test with the degree of freedom for this test and its *p*-value.

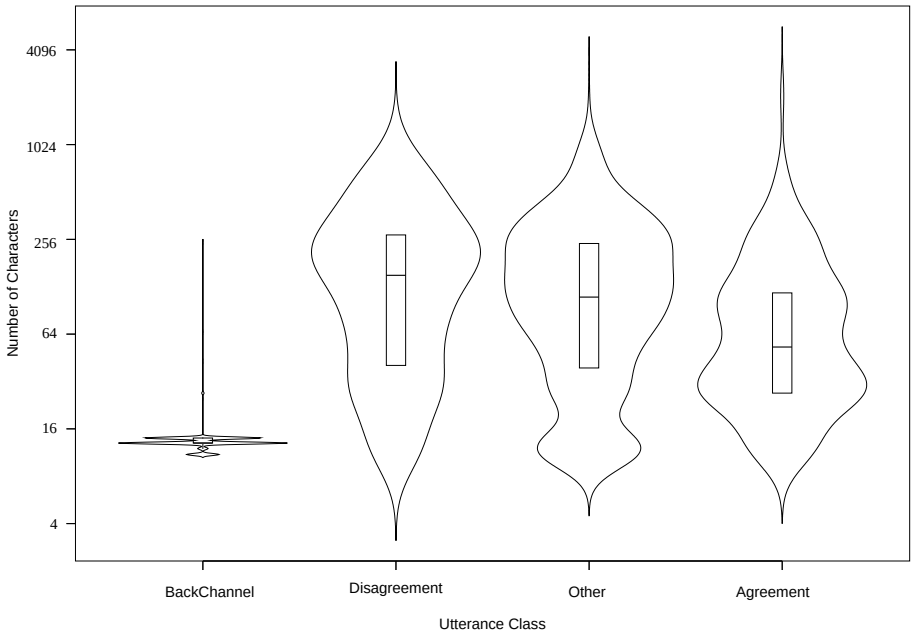

Figure 2. *Normalised density of the distribution of spurts according to their length feature for each class (log_2 scale). The distribution of spurts in the BackChannel class is skewed towards short spurts whereas the other classes spread along all the possible length. (The density is represented by the outer curves while the inside boxes show the lower and higher quartiles and the median)*

4.2. *First Word*

Hillard *et al.* (2003) use the "class" of the first word of the spurt as a feature. There is no real explanation in their article for the origin of this feature. However, this is a feature similar to the *discourse markers* theory (Schiffrin, 1988). In particular, Kotthoff (1993) discusses the set of specific disagreement markers, which are often found at the beginning of an utterance, for example:

– *disagreement downgrading markers* such as "well, I am afraid that [...]";

– *reduction of reluctancy markers* like "Yeah but".

By studying the annotated corpus, we can see that there is a significant difference between the first word vocabulary of each class. Table 1 shows the amount of overlap between each class vocabulary for the first word of their spurts.

Even if each class shares a number of identical first words with the other classes, their amount of use is significantly different. For example, Figure 3 shows the most frequently used first words and their distribution for each class. Even if a word is

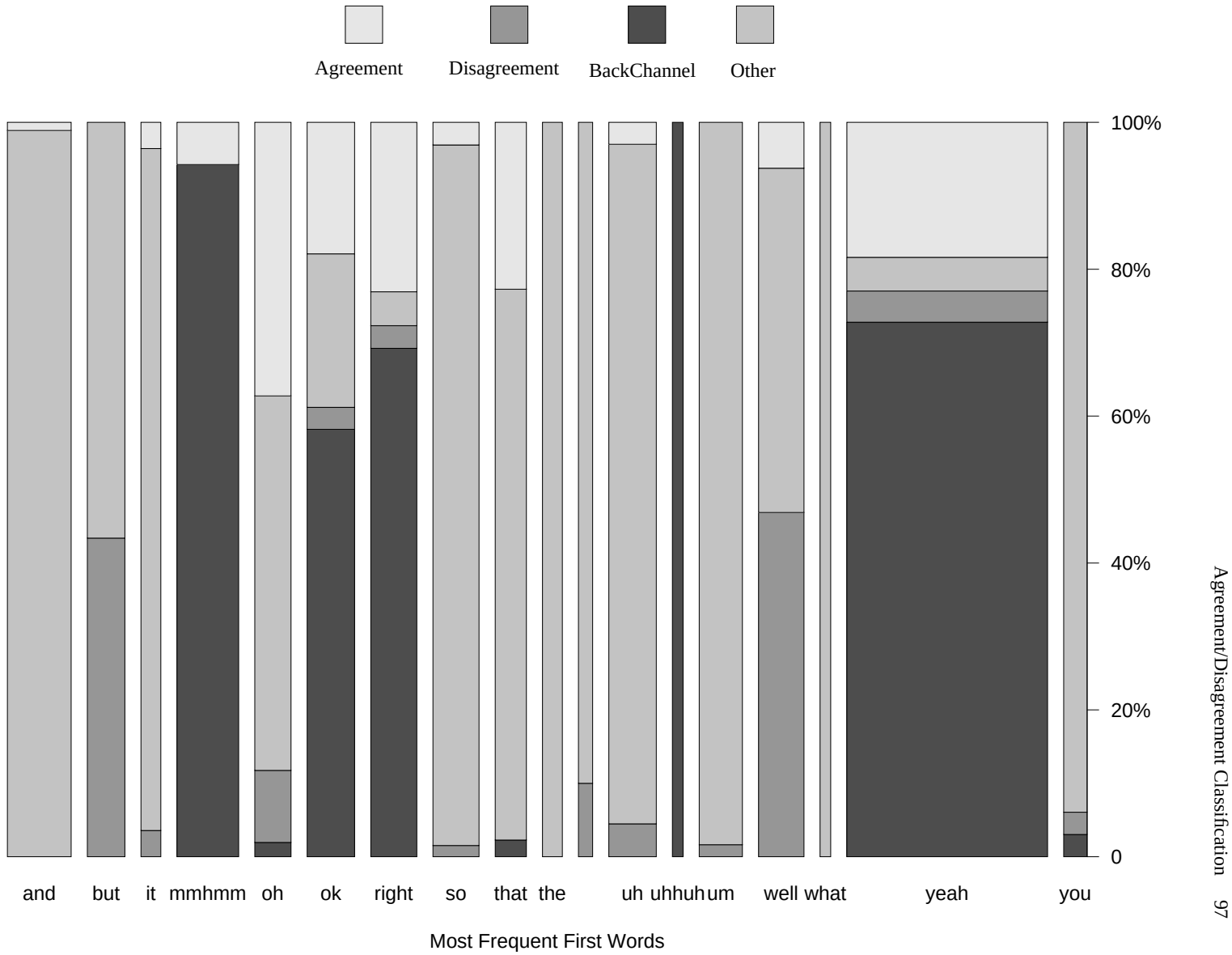

Figure 3. *Distribution between classes of the most frequent first words. The stacked vertical bars represent the number of utterances starting with the corresponding word in each class. The width of each bar represents the relative distribution of this word as a first word in the whole dataset. Most of the words are only found in a couple of classes with very different distribution, however some words are used as a first word of an utterance in all classes, like "yeah", but their use is often stronger in one class*

Disagreement	7.65%		
BackChannel	2.05%	1.47%	
Agreement	7.35%	3.53%	2.05%
	Other	Disagreement	BackChannel

Table 1. *Overlap of classes on the first word feature. The overlap is presented as a percentage of the size of the whole vocabulary of first words (340 words). The* Disagreement *and* Agreement *vocabularies overlap more with the* Other *vocabulary as this one is larger (271 words) due to the variance of the type of answers in the latter class*

found in each class as a first word, there is always a dominant class. For example, the majority of uses for "yeah" – which is the most frequent first word in the corpus – are in the *BackChannel* class. However, a few exceptions, such as "but" and "well", can be found where the use of the word is evenly distributed between two classes.

The first word feature is thus an interesting feature for discriminating between classes. In the proposed classifier (see Section 5) we use this feature directly instead of using the "class" of the word as Hillard *et al.* (2003) do.

4.3. *Punctuation*

As with discourse markers, punctuation appears to be a reasonable feature for the classification. However, again, there does not seem to be a strong theoretical literature on the linguistics of punctuation for Agreement and Disagreement utterances in dialogue.

We have conducted a statistical analysis of the annotated dataset and found three types of interesting punctuation: question marks, periods and commas. Exclamation marks do not display any significant difference in their use between classes. The feature we consider is the number of occurrences of a specific type of punctuation per utterance in a class.

We note that there are significantly more question marks ($M = 0.1$, $SD = 0.34$, $\hat{d} = 0.29$) in the utterance of class *Other* than in the rest of the utterances ($t(1776) = 5.06$, $p < 0.001$). A similar effect can be found for the other types of punctuation, as can be seen in figures 4 and 5.

5. Support Vector Machine Classifier

A hierarchical multi-class Support Vector Machine (SVM) classifier similar to the approach proposed by Vural and Dy (2004) is trained to obtain a multi-class labelling

of spurts. These classifiers are trained using the following shallow *local features* of the spurts:

– the length of the spurt (in characters);

– the first word of the spurt;

– spurts' bi-grams (i.e. all consecutive pairs of stemmed words in the spurt);

– part of speech tags (POS);

– number and type of punctuations in the spurt.

The POS and bi-grams features are standard features used in classification of text, which provide a generic view of the syntactic structure and semantic structure of a sentence without requiring too complex processing. The rest of the features are grounded in the discourse markers theory and empirical analysis of the annotated corpus as discussed in the previous section. A slightly different set of features is used, as we have added the Part of Speech tags, the bi-grams and the punctuations in the spurt to the "lexical" features proposed by Hillard *et al.* (2003) and Galley *et al.* (2004). The vocabulary features proposed in the previous state of the art have not been used as they did not show any significant influence on the classification results and can only be

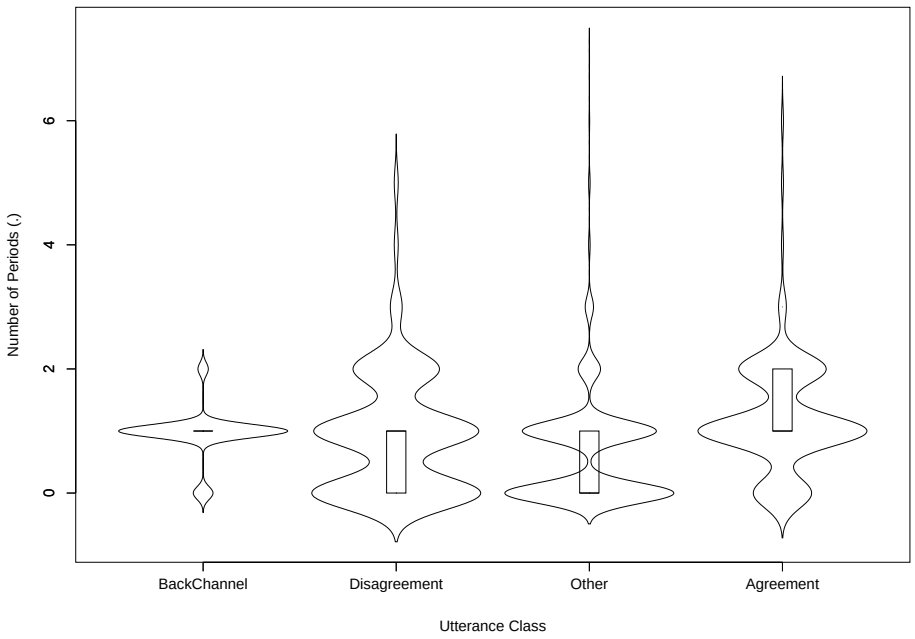

Figure 4. *Normalised density of use of periods (.) in each class. The probability of using a particular number of periods in an utterance is significantly different between each class*

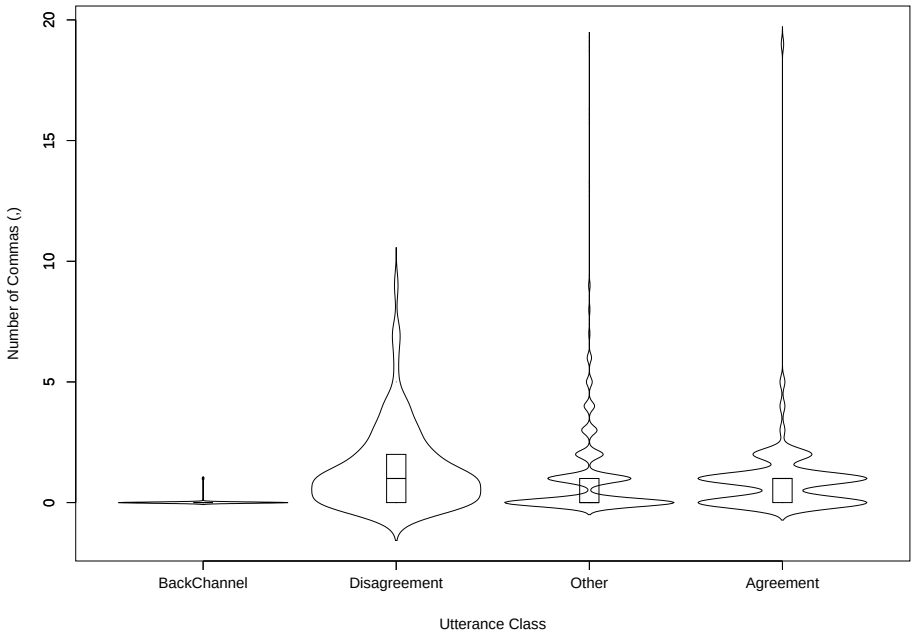

Figure 5. *Normalised density of use of commas (,) in each class. The probability of finding a particular number of commas in an utterance is significantly different between each class. For instance, there is little chance that a* BackChannel *will contain a comma, while a* Disagreement *utterance will most probably contain one or two periods*

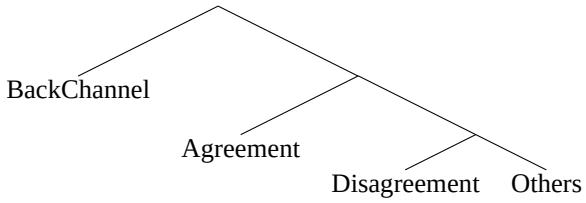

Figure 6. *Binary Support Vector Machine Classifiers in Cascade. Three binary SVM classifiers are used consecutively to label the spurts. The top classifier decides the* BackChannel *class; if the spurt is not of this class, the second classifier is applied to decide the* Agreement *class. If the spurt is not an* Agreement*, the last classifier is applied to choose between the* Disagreement *and* Others *classes*

tailor-built from the current corpus, raising the question of their coverage and applicability to other dialogues' vocabularies.

The Support Vector Machine classifier (Vapnik, 2000) can only be directly used for binary problems where we need to distinguish between two classes. There are different methods of combining a set of binary SVM classifiers to obtain a multi-class classifier. The most often used is the One-Versus-All combination where a set of classifiers is trained to classify one class against the rest of the classes. In our problem, we have four classifiers: Agreement versus All, Disagreement versus All, Other versus All and Disagreement versus All. In such a setup, a new utterance is classified by applying all classifiers and using the result that has the best score.

Vural and Dy (2004) show that the hierarchical combination of SVM can be as accurate as the standard One-Versus-All combination and can be usually be trained and tested faster. In our experiments, we have found (see Section 6) that the One-Versus-All classifier combination was not the best for the current task. We have thus used an alternative setup, a "cascade" of binary SVM, which can be compared to a decision tree where the branching decision is taken by an SVM (similar to the method described by Bennett and Blue (1998)). If there are N classes, this combination creates a cascade of classifiers where the root classifier compares 1 vs. N-1 classes, the second classifier compares 1 vs. N-2 classes, etc. In our task, the multi-class classifier is composed of 3 binary SVM classifiers [4] in cascade in the order BackChannel → Agreement → Disagreement vs. Others (see Figure 6). The cascade was built by selecting, at each node of the decision tree, the best performing binary classifier for the classes remaining at that level. Comparison between two different cascades is given in Section 6.

The rationale for the cascade classifier is that the difference between the BackChannel class and the other classes is easy to make based on the *length of the spurt* feature (see Section 4.1). After this first classification, there is less noise added by the BackChannel spurts in the learning. In a similar way, the *Agreement vs. Disagreement+Others* classification is mainly dependent on the *first word* feature, as agreement/disagreement spurts usually start with a limited set of words (e.g. "yes", "agree", "no", "well", etc.).

6. Results

In this section we report on the accuracy of our new model compared to the state of the art approaches and then analyse how the model could be improved to increase the overall accuracy. We discuss two different experimental setups that were used to evaluate the accuracy of the classifier in comparison with the existing setups in the state of the art.

4. The SVM is implemented with the Minorthird framework (Cohen, 2004).

6.1. *Setup 1*

The first setup reproduces Hillard *et al.*'s (2003) training/testing split across meetings. One meeting transcript is held out for testing and the classifier is trained on the rest of the meetings as a three-way classifier – *Agreement* and *BackChannel* classes being merged.

The split proposed by Hillard *et al.* (2003) is not random as it follows the corpus' split in individual meetings. Each meeting is different in topic and in participants and thus the content of each individual split might be biased toward a specific topic or a specific interlocutor's argumentation style. Training on one particular meeting to test on other meetings might thus produce results that are not representative of the general problem. We report the results on this setup to compare with previous work and discuss a second setup in the next section that uses a standard N-Fold cross-validation. However, the classifier model that we have described in the previous section performs as well in both setups.

Classifier	Error Rate	Error Rate Std. Dev.
BODA Cascade	13.53%	1.30%
BADO Cascade	**13.48%**	**0.96%**
One vs. All	17.78%	0.87%
Galley *et al.* (2004), Global Features	13.08 %	NA
Galley *et al.* (2004), Local Features	14.38%	NA
Hillard *et al.* (2003)	18%	NA

Table 2. Setup 1 *classifiers comparison. (Error rates for the state-of-the-art approaches are not available)*

The first setup was tested on three different binary classifiers combinations:

– a cascade: BackChannel → Others → Agreement vs. Disagreement (BODA);

– a cascade: BackChannel → Agreement → Disagreement vs. Others (BADO);

– a One-Versus-All SVM classifier.

The BackChannel → Others → Disagreement vs. Agreement cascade SVM (BODA) and BackChannel → Agreement → Disagreement vs. Others cascade SVM (BADO) achieve better results than the One vs. All SVM classifier (see Table 2). The cascade classifiers' accuracies are comparable to the state of the art techniques; in particular the accuracy is better than the classifier using only spurts features (*Local Features*) by Galley *et al.* and close to the classifier using adjacency pairs (*Global Features*).

6.2. *Setup 2*

The second setup performs a randomised five-fold cross-validation with the four-way classifier (comparable to the experimental setup used in Galley *et al.*, 2004). The 8,135 spurts are split randomly into five samples; each sample is consecutively used individually as a testing sample against a classifier trained on the rest of the samples.

Classifier	Error Rate	Error Rate Std. Dev.
BADO	**13.47%**	**0.57%**
Galley *et al.* (2004), Global Features	15.93%	NA
Galley *et al.* (2004), Local Features	16.89%	NA

Table 3. *Error rate of the classifiers for* Setup 2

Results from the second setup are reported by comparison to the state of the art techniques accuracies in Table 3. The accuracy of the *BackChannel* → *Agreement* → *Disagreement* vs. *Others* (BADO) SVM classifier is better than the state of the art classifiers while it only uses the spurt's local features. The BADO cascade performs slightly better in the second setup, but the difference is not significant and might be due to the more random distribution of features in the N-Fold than in the meeting split of Setup 1.

Galley *et al.*'s (2004) model slightly (0.4% [5]) outperforms our classifier in the first setup when using contextual features but does not seem to be as robust as the BODA when evaluated on the N-Fold random split. It is hard to explain why this is, as Galley *et al.* do not explain the cause of the difference in accuracies of their model between the two setups.

6.3. *Results Analysis*

To understand better what happens in the cascade of classifiers, we analysed the precision and recall for each individual class (as reported in Table 4) as well as the confusion between classes. Table 5 shows the number of spurts misclassified by the BADO cascade for each class and in which other class the spurt was wrongly classified.

The *Agreement* and *Disagreement* classes decrease the accuracy of the classifier (see Table 4) with an accuracy of 39% for the *Disagreement* class, while the *BackChannel* class has an accuracy of 98%. This is due to the small number of examples available in the corpus for the *Disagreement* and *Agreement* classes, with only 9.4% of spurts being instances of the *Agreement* class and 6.3% being instances of

5. No significance level can be computed as we do not have access to Galley *et al.*'s (2004) classification model and no standard deviation on their error rate is reported.

	BackChannel	Others	Agreement	Disagreement
Precision	**0.99**	0.90	0.67	0.38
Recall	**0.98**	0.91	0.62	0.40
F_1	**0.98**	0.91	0.64	0.39
Error Rate	**2.2%**	9.1%	37.8%	59.8%
Error Std. Dev.	1.5%	1.4%	9.9%	4.5%
Distribution in Corpus	22.6%	**61.7%**	9.4%	6.3%

Table 4. *Precision and Recall for Individual Classes in the BackChannel → Agreement → Disagreement vs. Others Cascade Classifier. The best results are in **bold** and the worst underlined*

	Predicted Classes			
Real Classes	BackChannel	Others	Agreement	Disagreement
BackChannel	**97.8%**	0.7%	1.2%	0.3%
Others	0.5%	**90.8%**	3.4%	5.3%
Agreement	4.7%	26.1%	**62.1%**	7.1%
Disagreement	0%	51.8%	8.0%	**40.2%**

Table 5. *Confusion Matrix for the* BackChannel → Agreement → Disagreement *vs.* Others *Cascade Classifier. This table presents the distribution of the annotated spurts by predicted classes. Each line represents a known class of spurt – from the corpus annotation – and the percentage of these spurts that were classified in another class. The spurts that were correctly predicted by the SVM classifier are in **bold**; the worst confusion of each line is underlined*

the *Disagreement* class. The *BackChannel* class, relying on the strong *length of spurt* feature, can be predicted easily, while the classification of the *Others* class, with 1,103 examples in the corpus (61.7% of the examples), can be trained with good accuracy.

In our application of this classifier within an argumentative dialogue manager, the classifier is applied to each of the user's utterances, trying to determine if the user is agreeing or disagreeing with the system.

The accuracy of the *Agreement* and *Disagreement* classes is individually low, however 51.8% of the misclassified *Disagreements* are labelled as *Others* (see Table 5) and there is little confusion between the *Agreement* and *Disagreement* classes themselves – the confusion between these two classes only accounts for 0.56% of the total error. As explained before, in the argumentative dialogue context the *Disagreement* and *Other* classes are merged and thus the confusion between these classes does not im-

pair the actual labelling of utterances as *Agreement* or *Disagreement* for this kind of task.

As discussed in Section 2, when users disagree with the system, they will try to defend their arguments with more support in the same topic. Even if there is a great confusion between the *Other* class and the *Disagreement* class, in this setup of an argumentative dialogue we can work around this problem by considering an utterance classified as *Other* as an invitation to continue discussing the same topic. We can thus process the *Disagreement* and the *Other* utterances in a similar manner in the dialogue manager. In this application, the confusion of these two classes is less influential.

7. Conclusion

A new approach is proposed to label the agreement of the user in dialogue utterances. The classification between *agreement* and *disagreement* utterances is based on a combination of binary Support Vector Machine (SVM) classifiers trained on a manually annotated corpus.

By using a SVM cascade, the classifier is able to achieve better results than the current state of the art approaches while using simpler features. Only local, shallow features such as the length of the utterances and their first word are used to determine the class of the utterance. The use of a cascade filters the *backchannel* utterances that are strongly characterised by the length of spurt feature, thus reducing the noise in the following classifiers.

Using a cascade reduces the noise in the lower-level classifiers and thus can improve their *precision*, but this could impair the *recall* of these classifiers. The results discussed in Section 6 show that this is not the case and that the precision and recall values are balanced.

To improve the accuracy of the classifier, the confusion between the *disagreement* class and the *others* class should be lowered. The difference between these two classes cannot be determined perfectly by shallow local features as they are very similar: *disagreement* and *others* utterances are both long and complex and do not always use strong discourse cues. The use of adjacency pairs, as proposed by Galley *et al.* (2004), could improve the classification by adding contextual information to the local features. In fact, if the classifier is used in the context of a dialogue management system, and one of the interlocutors is controlled by the system, adding *Adjacency Pairs* features and other contextual features – such as the type of answer expected by an utterance – might be easy to implement, as the system will know the pragmatic content of its own generated utterances without having to use deep processing.

This article has focused on the analysis of the relevant features for the classification of utterances in an argumentative dialogue management issue. More tailoring in the machine learning part of this model might also improve the classification accuracy; in particular Vural and Dy (2004) propose to tailor the kernel function of each binary

classifier in the decision tree to the particular classification problem to address the issue of uneven distribution of classes in the corpus.

8. References

Andernach T., "A Machine Learning Approach to the Classification of Dialogue Utterances", *CoRR*, 1996.

Andrews P., Manandhar S., De Boni M., "Argumentative Human Computer Dialogue for Automated Persuasion", *Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue*, Association for Computational Linguistics, Columbus, Ohio, pp. 138-147, June, 2008.

Bennett K., Blue J., "A support vector machine approach to decision trees", *The IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence*, 1998.

Cassell J., Bickmore T. W., "Negotiated collusion: Modeling social language and its relationship effects in intelligent agents", *User Modeling and Adaptive Interfaces*, vol. 13, No(s). 1-2, pp. 89-132, February, 2002.

Cohen W. W., "Minorthird: Methods for Identifying Names and Ontological Relations in Text using Heuristics for Inducing Regularities from Data", , web page `http://minorthird.sourceforge.net`, 2004.

Dinarelli M., Quarteroni S., Tonelli S., "Annotating spoken dialogs: from speech segments to dialog acts and frame semantics", *Proceedings of SRSL 2009 Workshop of EACL, Athens*, 2009.

Fernandez R., Picard R., "Dialog act classification from prosodic features using support vector machines", *Speech Prosody 2002, International Conference*, 2002.

Galley M., Mckeown K., Hirschberg J., Shriberg E., "Identifying agreement and disagreement in conversational speech: use of Bayesian networks to model pragmatic dependencies", *ACL '04: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Morristown, NJ, USA, 2004.

Gilbert M. A., Grasso F., Groarke L., Gurr C., Gerlofs J. M., "The Persuasion Machine", *in* C. Reed, T. J. Norman (eds), *Argumentation Machines: New Frontiers in Argument and Computation*, Springer, December, 2003.

Hillard D., Ostendorf M., Shriberg E., "Detection of agreement vs. disagreement in meetings: training with unlabeled data", *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, Association for Computational Linguistics, Morristown, NJ, USA, pp. 34-36, 2003.

Jackson S., Jacobs S., "Structure of Conversational Argument: Pragmatic Bases for the Enthymeme", *Quarterly Journal of Speech*, vol. 66, No(s). 3, pp. 251-265, 1980.

Janin A., Baron D., Edwards J., Ellis D., Gelbart D., Morgan N., Peskin B., Pfau T., Shriberg E., , Stolcke A., Wooters C., "The ICSI Meeting Corpus", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 364-367, April, 2003.

Kotthoff H., "Disagreement and Concession in Disputes: On the Context Sensitivity of Preference Structures", *Language in Society*, vol. 22, No(s). 2, pp. 193-216, 1993.

Levin J. A., Moore J. A., “Dialogue-games: metacommunication structures for natural language interaction”, *Cognitive Science*, vol. 1, No(s). 4, pp. 395-420, 1977.

Mazzotta I., de Rosis F., Carofiglio V., “Portia: A User-Adapted Persuasion System in the Healthy-Eating Domain”, *Intelligent Systems, IEEE*, vol. 22, No(s). 6, pp. 42-51, 2007.

Schiffrin D., *Discourse Markers (Studies in Interactional Sociolinguistics)*, Cambridge University Press, February, 1988.

Stolcke A., Ries K., Coccaro N., Shriberg E., “Dialogue act modeling for automatic tagging and recognition of conversational speech”, *Computational linguistics*, 2000.

Stolcke A., Shriberg E., Bates R., Coccaro N., “Dialog act modeling for conversational speech”, *AAAI Spring Symposium on Applying Machine Learning to Discourse Processing*, 1998.

Vapnik V. N., *The Nature of Statistical Learning Theory*, Springer, 2000.

Vural V., Dy J. G., “A hierarchical method for multi-class support vector machines”, *ACM International Conference Proceeding Series*, vol. 69, ACM, 2004.