D4.3 – The SENSEI Discourse Analysis Tools, 3

Document Number	D4.3
Document Title	The SENSEI Discourse Analysis Tools, 3
Version	1.0
Status	Final
Workpackage	WP4
Deliverable Type	Report
Contractual Date of Delivery	31.10.2016
Actual Date of Delivery	31.10.2016
Responsible Unit	UESSEX
Keyword List	discourse parsing, event/temporal structure, argumentation structure, intra/inter document coreference
Dissemination level	PU

Editor

Mijail Kabadjov	(University of Essex, UESSEX)
Evgeny A. Stepanov	(University of Trento, UNITN)

Contributors

Evgeny A. Stepanov	(University of Trento, UNITN)
Fabio Celli	(University of Trento, UNITN)
Shammur A. Chowdhury	(University of Trento, UNITN)
Ahmet Aker	(University of Sheffield, USFD)
Adam Funk	(University of Sheffield, USFD)
Mijail Kabadjov	(University of Essex, UESSEX)
Udo Kruschwitz	(University of Essex, UESSEX)
Massimo Poesio	(University of Essex, UESSEX)

SENSEI Coordinator

Prof. Giuseppe Riccardi
Department of Information Engineering and Computer Science
University of Trento, Italy
giuseppe.riccardi@unitn.it

Document change history

Version	Date	Status	Author (Unit)	Description
0.1	15/07/2016	Draft	M. Kabadjov, M. Poesio, U. Kruschwitz (UESSEX)	Outline.
0.2	20/08/2016	Draft	M. Kabadjov (UESSEX)	Added section 4 (task 4.3).
0.3	25/08/2016	Draft	F. Celli (UNITN)	Added section 5 (task 4.5).
0.4	30/08/2016	Draft	E.A. Stepanov (UNITN)	Modified Outline.
0.4	30/08/2016	Draft	S. Chowdhury (UNITN)	Added Section 2.1 (task 4.1).
0.4	30/08/2016	Draft	E.A. Stepanov (UNITN)	Added Section 2.2 (task 4.1).
0.5	31/08/2016	Draft	E.A. Stepanov (UNITN)	Added Section 2.3 (task 4.1).
0.6	7/09/2016	Draft	M. Kabadjov (UESSEX)	Updated section 4 (task 4.3).
0.7	7/09/2016	Draft	E.A. Stepanov & F. Celli (UNITN), M. Kabadjov (UESSEX)	Intro and Conclusions.
0.8	10/09/2016	Draft	A. Aker (USFD), M. Kabadjov (UESSEX)	Added clustering in section 4 (task 4.3).
0.9	19/09/2016	Draft	A. Aker (USFD)	Internal Scientific Review.
0.10	20/09/2016	Draft	E. Chiarini (UNITN)	Quality Check.
0.11	24/09/2016	Draft	E.A. Stepanov (UNITN), M. Kabadjov (UESSEX)	Addressed scientific review and quality check points.
0.12	05/10/2016	Draft	A. Funk (USFD)	Added section on T4.2.
0.13	05/10/2016	Draft	A. Funk (USFD)	Updated executive summary, follow-up to Period 2, conclusion.
0.14	07/10/2016	Draft	E. Chiarini (UNITN)	Final Check.
0.15	07/10/2016	Draft	E.A. Stepanov (UNITN), M. Kabadjov (UESSEX)	Final version for submission.
1.0	07/10/2016	Final	G. Riccardi (UNITN)	Approval for submission.

Executive summary

In this deliverable we present the progress on the discourse analysis methods developed within the project in Period 3. We continued the lines of work pursued during Period 2 of the project on discourse parsing of spoken conversations, on inter- and intra-document coreference in social media, and on argument structure of conversations started in Period 2.

The document is organised as follows: in Section 2, progress on discourse parsing for conversations is presented. Next, progress in event, temporal expression, and sentiment detection is described (Section 3), followed by progress on intra- and inter-document coreference resolution for conversations in social media (Sections 4 and 5, respectively), and finally the work on argument structure is discussed in Section 6. Finally, conclusions and future plans are drawn.

Table of Contents

List of Acronyms and Abbreviations

Acronym	Meaning
AMT	Amazon Mechanical Turk
HGI	Harvard General Inquirer lexicon
JRC	Joint Research Centre of the European Commission
MLP	Multi-layer Perceptron
MPQA	Multi-Perspective Question Answering corpus/lexicon
NE	Named Entity
NLP	Natural Language Processing
NN	Neural Network
OnForumS	Online Forum Summarisation
PoS/POS	Part-of-Speech
SIGdial	Special Interest Group on Discourse and Dialogue
SVM	Support Vector Machines
MXE	Maximum Entropy
UWB	University of West Bohemia
WP	Work Package

1 Introduction

The objective of WP4 is to develop tools supporting automated discourse analysis of conversations both as happening online (e.g., online forums) as well as in spoken dialogue (e.g., customer call centres). In particular, we aim to develop tools for discourse parsing, event/temporal structure, argumentation structure, and intra-/inter-document coreference in the two domains (social media conversations and call centre conversations) and three languages (English, French, and Italian) of the project. A key goal of the research is to investigate the performance of techniques developed for the most extensively studied forms of language use (e.g., news) in these new domains, and develop methods for adapting such techniques.

The objectives for Period 3 were to develop the third and final release of these tools continuing with the emphasis on methods for domain adaptation and on the creation of appropriate resources for enabling this work.

In Deliverable D4.3 we present progress and completion of Work Package 4 during Period 3. It provides details on the tools used for the tasks of the work package, the experiments carried out on developing and adapting these tools to the domains of interest to the project, and plans for future work along the various lines of work beyond the end of the project.

1.1 Follow-up to Period 2 Activities

During Period 3 of the project, on Discourse Parsing we have implemented automatic segmentation of dialog acts with respect to dimensions defined by the ISO 24617-2 standard. Additionally, we have improved end-to-end PDTB-style discourse parsing system. Full details on this can be found in Section 2).

The event and sentiment detection tools were expanded to include temporal expression detection and to use a newer and better sentiment detection component, as described in Section 3. (The new component is not freely licenced for non-academic use, so the old one is still available in the combined tool as an option.)

During Period 3, work on adapting the Coreference Resolution Toolkit, BART, continued and there were two main developments: one, a new corpus created by USFD was prepared with thread structure and annotated with coreferences at UESSEX, and another, a new set of machine learning features were devised to model thread structure and incorporated into a new thread-aware version of BART. We evaluate intrinsically, as well as, extrinsically the effects of Coreference on Clustering. Additionally, work on integrating BART's output with the repository (cf. D5.1) was also carried out. Full details on this line of work can be found in Section 4.

During Period 3, taking advantage of the newly created USFD corpus, we repeated the experiments ran in Period 2 using the JRC-Names resource [49] on the new dataset and we report the results. This line of work is described in Section 5.

In the follow-up of Period 2, on Task 4.5 we improved the performance of the agreement/disagreement structure labelling algorithm, and also improved its integration into the template-based summary (reported in D5.3). In addition to that, work took place on preparing a data set for a second edition of the shared task ONFORUMS planned for 2017. This work is presented in Section 6.

2 Task 4.1: Discourse parsing for conversations

In the deliverable D4.2 we have adopted the latest instantiation of the ‘information state update’ model given by the ISO 24617-2 international standard on dialogue act annotation, and defined the Task 4.1 activities with respect to this model. According to the standard, a conversation consists of several functional segments – minimal spans of behavior (verbal or not) that have a communicative function. Communicative functions take place across multiple semantic dimensions (segments are dimension specific and can overlap). Thus, in a model, a dialogue act consists of a communicative function – semantic dimension pair, such that some communicative functions are dimension specific and others are general. A dialogue act has several participants: at least one sender and one or more addressees. A communicative function can be described by function qualifiers for aspects such as sentiment, certainty, and conditionality. Dialogue acts can be connected to each other by functional and feedback dependency relations and rhetorical/discourse relations. Discourse relations additionally connect semantic content to other dialogue acts or semantic content units of a conversation.

In Period 2 of the project we addressed task of dialog act classification on Italian LUNA corpus and identified dimensions and communicative functions of dialogue segments. As in the ISO definition a dialog turn can consist of several dialog acts, in Period 3 we have addressed the task of automatic segmentation of utterances into dialog acts. The activity is described in Section 2.1.

In Section 2.2 we present the improvements on discourse parser developed within Period 2 of SENSEI. The parser was submitted for participation in CoNLL 2016 Shared Task on Shallow Discourse Parsing and in the end-to-end evaluation ranked 3rd (out of 14 participating systems).

2.1 Dialogue Act Segmentation

For the complete automation of the detection and classification of dialog acts with respect to DiaML specification, a final set of experiments involved designing an automated segmenter which takes the dialog turns as an input. The automated segmenter extracts token/word with a context of ± 2 as feature and uses a discriminative approach, namely Conditional Random Fields (CRFs), to simultaneously segment an utterance into its DA boundaries, in IOB format. The performance of the system is shown in Table 1.

The segmented DA boundaries are then passed through the DA classifier, described in Deliverable D4.2 and label such segments according to a DA tag. The results of the pipeline with error propagation – automatic segmentation and classification – are presented in Table 2.

As for evaluation we reported both exact and partial match performances. The partial match computes the intersection of the exact match and the overlap between reference and hypotheses [24]. It is useful for cases when an exact alignment is not necessary.

Since the Dialog Acts we are interested in are of *general* dimension, which achieves F-measure of 0.61 for exact matches and 0.81 for partial matches, the performances are acceptable for downstream applications.

Table 1: Precision (P), recall (R) and F1 of dialogue act segmentation

Segmenter	**P**	**R**	**F1**
Overall	0.73	0.59	0.65

Table 2: Precision (P), recall (R) and F1 of dialogue act segmentation followed by dialog act classification

Segmenter+DA Classification	**P**	**R**	**F1**
Exact - Overall	***0.46***	***0.58***	***0.51***
General	0.59	0.67	0.63
Social	0.33	0.44	0.38
Time+Feedback	0.29	0.43	0.34
Other	0.12	0.18	0.15
Partial -Overall	**0.62**	**0.72**	**0.67**
General	0.79	0.83	0.81
Social	0.52	0.69	0.59
Time+Feedback	0.36	0.55	0.43
Other	0.17	0.21	0.19

2.2 PDTB-Style Discourse Parsing

Penn Discourse Treebank style discourse parsing is a composite task of detecting explicit and non-explicit discourse relations, their connective and argument spans, and assigning a sense to these relations. In this section we describe the end-to-end discourse parser for English. In Period 3, the main focus of the parser was on argument spans and the reduction of global error through model selection.

2.2.1 System Architecture

The discourse parser is the modified version of the parser developed by [52] and described in the deliverable D4.2. The system is an extension of the explicit relation parser described

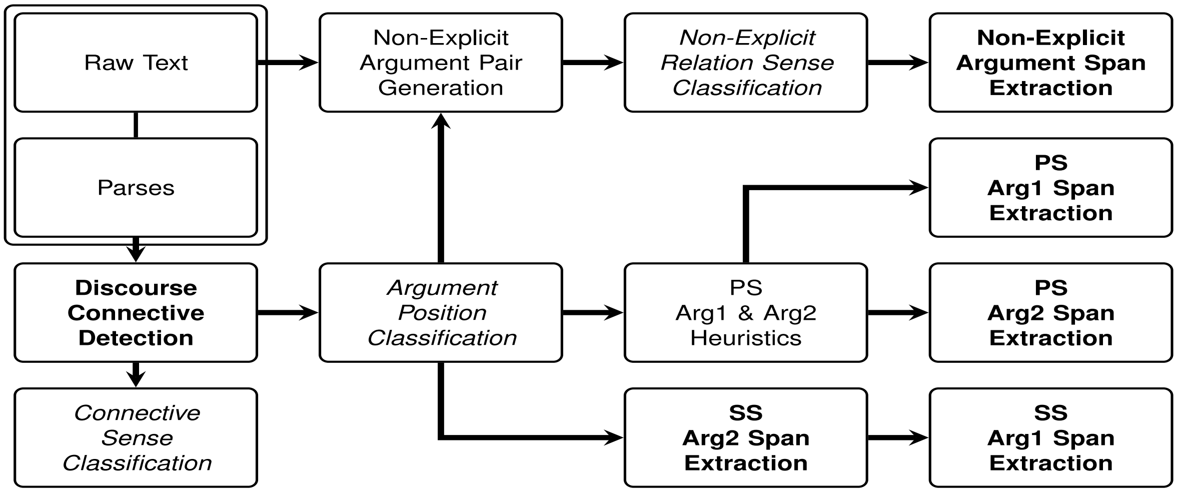

Raw Text
Parses
Non-Explicit Argument Pair Generation
Non-Explicit Relation Sense Classification
Non-Explicit Argument Span Extraction
PS Arg1 Span Extraction
Discourse Connective Detection
Argument Position Classification
PS Arg1 & Arg2 Heuristics
PS Arg2 Span Extraction
Connective Sense Classification
SS Arg2 Span Extraction
SS Arg1 Span Extraction

Figure 1: Discourse parsing architecture: the sequence labeling modules are in **bold** and the classification modules are in *italics*.

in [50; 51]. The overall architecture of the parser is depicted in Figure 1. The approach implements discourse parsing as a pipeline of several tasks such that connective and argument span decisions are cast as sequence labeling and sense decisions as classification.

The discourse parsing pipelines starts with the identification of discourse connectives and their spans (*Discourse Connective Detection* (DCD)), and is followed by *Connective Sense Classification* (CSC) and *Argument Position Classification* (APC) steps. While CSC assigns sense to explicit discourse relations, APC classifies them as intra- and inter-sentential (*Same Sentence* (SS) and *Previous Sentence* (PS) *Argument 1*). Both tasks operate using the connective span tokens only.

With respect to the decision of the *Argument Position Classification* the pipeline is split into explicit and non-explicit tasks. For the explicit relations, specific *Argument Span Extraction* (ASE) models are applied for each of the arguments with respect to their begin intra- or inter-sentential. Since *Argument 2* is syntactically attached to the discourse connective, its identification is easier. Thus, for the intra-sentential (SS) relations, models are applied in a cascade such that the output of *Argument 2* span extraction is the input for *Argument 1* span extraction. For the inter-sentential (PS) relations, on the other hand, a sentence containing the connective is selected as *Argument 2*, and the sentence immediately preceding it as a candidate for *Argument 1*.

For non-explicit discourse relations, a set of candidate argument pairs is constructed using adjacent sentence pairs within a paragraph and removing all the sentence pairs already identified as inter-sentential explicit relations. Each of these argument pairs is assigned a sense using *Non-Explicit Relation Sense Classification* (NE-RSC) models and their argument spans are extracted using *Non-Explicit Argument Span Extraction* step.

In the discourse parser, the *Non-Explicit Relation Sense Classification*, *Connective Sense Classification*, and *Argument Position Classification* tasks are cast as supervised classification using AdaBoost algorithm [20] implemented in *icsiboost* [18]. The span extraction tasks (*Discourse Connective Detection* and explicit and non-explicit *Argument Span Extraction*), on the other hand, are cast as token-level sequence labeling with CRFs [29] using CRF++ [28]. Besides training the CRF models for ASE, for inter-sentential *Argument 1* span and both non-explicit argument spans, we also make use of the 'heuristics': taking an argument sentence as a whole and removing leading and trailing punctuation [30; 52]. In the next section we describe the features used for the tasks.

2.2.2 Features

All the discourse parsing sub-tasks (both classification and sequence labeling) except *Non-Explicit Relation Sense Classification* make use of token-level features. However, the feature sets for each task are different. Compared to the version of the parser developed within Period 2, the feature sets are different; thus, additional descriptions are provided together with the old ones.

Token-Level Features Table 3 gives an overview of feature sets per task. Besides tokens and POS-tags, the rest of the features are described below.

Chunk-tag is the syntactic chunk prefixed with the information whether a token is at the beginning (B-), inside (I-) or outside (O) of the constituent (i.e. IOB format) (e.g. 'B-NP' indicates that a token is at the beginning of Noun Phrase chunk). The information is extracted from constituency parse trees using chunklink script [10].

IOB-chain is the path string of the syntactic tree nodes from the root node to the token, similar to Chunk-tag, it is prefixed with the IOB information. For example, the IOB-chain 'I-S/B-VP' indicates that a token is the first word of the verb phrase (B-VP) of the main clause (I-S).The feature is also extracted using the chunklink script [10].

Dependency chain [52] is a feature inspired by *IOB-chain* and is the path string of the functions of the parents of a token, starting from the root of a dependency parse.

VerbNet Class [27] is a feature intended to capture attributions. The feature requires lemmas, which were extracted using TreeTagger [48].

Connective Label and *Argument 2 Label* are the output labels of the *Discourse Connective Detection* and *Argument 2 Span Extraction* models respectively.

Using templates of CRF++ the token-level features are enriched with ngrams (2 & 3-grams) in the window of ± 2 tokens, such that for each token there are 12 features per feature type: 5 unigrams, 4 bigrams and 3 trigrams. All features are conditioned on the output label inde-

Feature	DCD	CSC	APC	ASE: SS		ASE: PS		NE-ASE	
				A1	*A2*	*A1*	*A2*	*A1*	*A2*
Token	Y	Y	Y	Y	Y	Y	Y	Y	Y
POS-tag	Y		Y		Y	Y	Y	Y	Y
Chunk-tag	Y								
IOB-chain	Y		Y	Y	Y	Y	Y	Y	Y
Dependency chain					Y				
VerbNet class					Y				
Connective Label				Y	Y		Y		
Argument 2 Label				Y					

Table 3: Token-level features for classification and sequence labeling tasks: Discourse Connective Detection (DCD), Connective Sense Classification (CSC), Argument Position Classification (APC), and Argument Span Extraction (ASE) of intra- (SS) and inter-sentential (PS) explicit and non-explicit (NE) relations.

pendently of each other. Additionally, CRFs consider the previous token's output label as a feature.

Argument and Relation-level Features In this section we describe the features used for *Non-Explicit Relation Sense Classification*. Previous work on the task makes use of a wide range of features; however, due to the low state-of-the-art on the task, we focused on the following features: sentiment polarities from MPQA lexicon [61], Brown Clusters [55], and VerbNet [27]. Similar to VerbNet Class feature, described above, lemmas from TreeTagger [48] are used to compute the polarity features.

There are four features generated for *Polarity*: (1-2) Individual argument polarities computed from token-level polarities as a difference of counts of positive and negative polarity words. The feature is assigned either 'negative' or 'positive' value with respect to the difference. (3) The concatenation of the argument polarity values (e.g. *negative-positive*). (4) The boolean feature indicating whether the argument polarities match.

The Brown Cluster and VerbNet features are extracted only for specific tokens. Starting from the dependency parse trees of the arguments we extract the main verb (root), subject (including passive), direct and indirect objects for each of them. Since for extracting VerbNet features we make use of lemmas, the lemmas themselves are considered for classification as well. Similar to polarity, the *VerbNet* features (4) are main-verbs' classes of the arguments, their concatenation, and a boolean feature indicating their match.

The *Brown Cluster* and *Lemma* features are main-verbs' brown clusters and lemmas, their concatenation and boolean features for matches (4). Unlike VerbNet, these features are also generated for a Cartesian product for the arguments' subject, direct and indirect objects. Consequently, there are 4 features for verbs and 24 for other dependency roles (3 + 3 + 9 + 9) per feature type.

2.2.3 Individual Modules

In this section we provide implementation details for the individual components of the discourse parser. We first address explicit and then non-explicit relations.

Explicit Discourse Relations

The explicit relation pipeline consists of *Discourse Connective Detection*, *Connective Sense Classification*, *Argument Position Classification* and *Argument Span Extraction* tasks. *Connective Sense Classification* and *Argument Position Classification* components did not undergo any change from Period 2; the other components are described below.

Discourse Connective Detection Since *Discourse Connective Detection* is the first step in discourse parsing, the performance of the task is critical. The task is cast as sequence labeling with CRFs. The performance of the models is tuned by feature ablation to yield a model that achieves F_1 of 0.9332 on the development set. The best model is trained on cased tokens, POS-tags, Chunk-tag and IOB-chain features.

Argument Span Extraction *Argument Span Extraction* is the main focus of the development. We train CRF model for each of the arguments of the intra- and inter-sentential relations considering a single sentence as a candidate (i.e. all multi-sentence relations are missed). As a candidate for the inter-sentential *Argument 1* we consider only immediately preceding sentence (effectively missing all non-adjacent *Argument 1* relations).

Since *Argument 2* models make use of connective span labels as a feature, and intra-sentential *Argument 1* model makes use of both connective and *Argument 2* labels; these models are trained using reference annotation spans. For the *Argument Span Extraction* of inter-sentential *Argument 1*, additional to the training of the CRF models we also make use of the heuristic, that takes the sentence as a whole and removes leading and trailing punctuation.

There are 4 CRF models for the task with the additional heuristic for the inter-sentential *Argument 1*. The feature sets for each of the models are selected such that they maximize the F-measure of both arguments together.

The CRF model for the inter-sentential *Argument 1* yields higher performance than the heuristic. However, the submitted system exploits the heuristic, since the difference between the two for the both argument spans is not large (0.4981 vs. 0.4936 for the heuristic).

Task	Explicit			Non-Explicit			All Relations		
	Dev	Test	Blind	Dev	Test	Blind	Dev	Test	Blind
Connective	0.9332	0.9243	0.8856	–	–	–	0.9332	0.9243	0.8856
Arg 1	0.5566	0.4964	0.5028	0.6951	0.6558	0.6683	0.6417	0.5890	0.5991
Arg 2	0.7907	0.7651	0.7205	0.7451	0.6778	0.7911	0.7664	0.7188	0.7586
Arg 1+2	0.4936	0.4456	0.4184	0.5940	0.5180	0.5805	0.5471	0.4844	0.5060
Parser	0.4589	0.3960	0.3174	0.2089	0.1756	0.1946	0.3246	0.2780	0.2510

Table 4: Task-level and end-to-end F_1-measures of the discourse parser on the development, test, and blind test sets of CoNLL 2015/2016 Shared Task for explicit and non-explicit relations individually and jointly for all relations. The task-level performances are reported with the error propagation. Thus, the *sense classification* performances are equivalent to the end-to-end parser performances.

Non-Explicit Discourse Relations

The non-explicit relation parsing pipeline consists of *Relation Sense Classification* (NE-RSC) and *Argument Span Extraction* (NE-ASE) tasks. Even though, NE-ASE is applied after NE-RSC with the idea of exploiting classification confidences for filtering out the candidate relations, the two tasks are fairly independent.

Non-Explicit Relation Sense Classification The set of features for the task is described in Section 2.2.2. It is the only task that makes use of the argument and relation level features. Due to the low state-of-the-art on the task, the focus is on the development of the models that maximize the performance of the majority senses – *EntRel* and *Expansion.Conjunction*. The flat classification mode is considered as it yields higher performance for these senses (e.g. for EntRel the classification into 4 top-level senses + EntRel yields F_1 of ≈ 0.30, while flat classification into 14 full senses + EntRel F_1 of 0.44).

Non-Explicit Argument Span Extraction The task is implemented similar to the *Argument Span Extraction* of the inter-sentential *Argument 1*, and considers the same feature set (cased token, POS-tag, and IOB-chain). Similarly, we experiment with the span extraction heuristic by only removing leading and trailing punctuation.

Unlike explicit relations, the CRF models for the non-explicit argument span extraction perform significantly better than the heuristics. However, due to the error propagation from the Relation Sense Classification task, the heuristics yield the higher F_1-measure for the end-to-end parsing of non-explicit relations. Thus, the system contains purely heuristic *Non-Explicit Argument Span Extraction*.

System	P	R	F
Period 3	**0.2622**	**0.2407**	**0.2510**
Period 2	0.2094	0.2283	0.2184

Table 5: Precision (**P**), recall (**R**) and F_1 (**F**) of the end-to-end discourse parsing on the blind test set for the Period 2 and Period 3 parsers.

2.2.4 End-to-End Parsing and Sub-component Performances

The end-to-end parsing performance is evaluated within CoNLL 2016 Shared Task on Shallow Discourse Parsing on a per-discourse relation basis on a PDTB [42] development and test sets (sections 22 and 23, respectively), and a blind test set specifically annotated for the shared task. For the evaluation, a relation is considered to be predicted correctly only in case the parser correctly predicts (1) discourse connective head, (2) exact spans and labels of both arguments, and (3) sense of a relation. The reported evaluation metrics are (1) explicit discourse connective, (2-4) Argument 1 and Argument 2 spans individually and together, and the sense of a relation. The reported micro-F1 measure of the sense classification is equivalent to the end-to-end parsing performance as it considers the error propagation from the upstream tasks. The metrics are reported for explicit and non-explicit relations individually and jointly in Table 4. The system achieves end-to-end parsing F1 of 0.3246, 0.2789 and 0.2510 on the development, test and blind test sets respectively.

2.2.5 Comparison to Period 2 System

We first compare the system performance to the Period 2 system on the end-to-end parsing score on the blind test set (see Table 5). The current system outperforms the Period 3 system on all the metrics.

The major change from Period 2 is the elimination of the *Non-Explicit Relation Detection* step. The step classified non-explicit relation candidates into relations and non-relations. However, the ratio of non-related adjacent sentence pairs usually is very low (circa 1%). Consequently, the step was penalizing the performance on non-explicit relations. As it can be observed from Table 6, there is a major improvement in performance for non-explicit argument spans.

The other changes are in the feature sets of *Connective Detection* and the *Argument Span Extraction* of the explicit intra-sentential *Argument 2*. For the former we improved the performance on the development set, but the performance on the test and blind test sets dropped (see Table 7). For the latter, we introduced a new feature – VerbNet [27] classes – intended to capture the attribution spans. From the results it appears that the feature is useful, as they are better despite the lower connective detection performance.

System	**Dev**	**Test**	**Blind**
Arg 1+2 Span Extraction			
Period 3	**0.5940**	**0.5180**	**0.5805**
Period 2	0.4000	0.3730	0.3831
Non-Explicit Parsing			
Period 3	**0.2089**	**0.1756**	**0.1946**
Period 2	0.1577	0.1330	0.1577

Table 6: F_1 for the non-explicit argument extraction and parsing.

System	**Dev**	**Test**	**Blind**
Discourse Connective Detection			
Period 3	**0.9332**	0.9243	0.8856
Period 2	0.9219	**0.9271**	**0.8992**
Explicit SS Arg 2			
Period 3	**0.7907**	**0.7651**	**0.7205**
Period 2	0.7748	0.7616	0.7068

Table 7: F_1 for the *Discourse Connective Detection* and explicit intra-sentential *Argument 2* span extraction.

2.3 Discourse Connective Detection in Spoken Conversations

Mainly due to the lack of discourse annotated dialog data, most of the research on discourse parsing has focused on written text; and discourse parsers heavily rely on features extracted from syntactic parse trees (e.g. [30; 50; 52]). Unfortunately, syntactic parsers trained on written text behave poorly on dialog data [9], since the latter contain disfluencies and no sentence segmentation. In this section we present experiments on discourse connective detection – initial step in Penn Discourse Treebank (PDTB) [42] style discourse parsing – in Italian spoken dialogues using acoustic and lexical features.

Detection of discourse connectives from English text using syntactic features has a very high performance ($F_1 = 94.19$) [37]. Detection of discourse connectives from spoken dialogues is more challenging than from written text. In addition to the coordination of non-discourse units and polysemy, which occur both in dialogues and written text, in spontaneous conversations words that can function as discourse connectives can also function as *discourse markers* [47]. While discourse connectives relate discourse units, discourse markers are used for discourse organization and turn management, e.g. *allora* (so) in Example (1) is a discourse connective and in (2) it is a discourse marker [54]. Our goal is to discriminate between discourse connective category of a word token and all other usages.

(1) [*In questo momento il palazzo non è collegato*]$_{Arg1}$
Allora [**è meglio collegarlo**]$_{Arg2}$
([*At this moment the building is not connected*]$_{Arg1}$
So [**we'd better connect it**]$_{Arg2}$)

(2) Allora vediamo un po' ecco qua
(So let's see here it is)

Figure 2: Examples of *allora* (*so*) used as a discourse connective (1) and as a discourse marker (2).

We cast discourse connective detection as a binary classification task using lexical and acoustic features. We focus on the 10 most frequent Italian discourse connectives in the LUNA corpus [16] of human-human spoken conversations. Since our goal is to explore the relevance of acoustic and lexical context for the task, we experiment with features extracted from connective candidate spans and their left and right contexts in the window of ± 2 tokens. We observe that both lexical and acoustic context have mixed effect on the detection of specific connectives.

2.3.1 LUNA Discourse Annotation

A subset of 60 dialogues from Italian LUNA Human-Human Corpus [16] was annotated [54] for discourse relations following Penn Discourse Treebank (PDTB) [42] guidelines. Out of total 1,606 annotated discourse relations, 1,052 are *explicit* discourse relations, that are signaled by 85 unique discourse connectives. Here we focus only on the 10 most frequent ones that are listed in Table 8 with their frequencies in data (Data Freq. column). This set of connectives accounts for 75.7% of all annotated explicit discourse relations. Some of the listed connectives additionally occur as tokens of other multi-word connectives (e.g. *che* is part of *visto che*). The amount of such multi-word connectives is 6.2%, most frequent being *che* (4.5%). To reduce noise, these multi-word connectives are removed from data.

The data (60 dialogues) is split into training, development and test splits as 42, 6, and 12 dialogues respectively. The distribution of the selected connectives into training and test sets after data pre-processing is given in Table 9.

2.3.2 Feature Extraction

In this section we first describe data pre-processing; then feature extraction and the features themselves.

Connective		Data Freq.		ASR Freq.	
e	(and)	160	15.2%	154	96.2%
perchè	(because)	138	13.1%	136	98.6%
allora	(so)	91	8.7%	87	95.6%
però	(but)	87	8.3%	86	98.9%
ma	(but)	71	6.7%	71	100%
quindi	(then/so)	69	6.6%	67	97.1%
poi	(then)	62	5.9%	59	95.3%
se	(if)	60	5.7%	58	96.7%
così	(so)	33	3.1%	31	93.9%
che	(that)	25	2.4%	23	92.0%
Top 10		796	75.7%	772	96.3%
Rest (75)		256	24.3%		
Total (85)		1,052	100%		

Table 8: The 10 most frequent connectives in the LUNA Corpus and their % from total of *explicit* relations. ASR Freq. column gives % of connectives recognized by ASR.

Word	02: Train				03: Test			
	CONN		O		CONN		O	
e	97	43.3%	127	56.7%	40	46.0%	47	54.0%
perchè	81	83.5%	16	16.5%	35	83.3%	7	16.7%
allora	61	16.3%	313	83.7%	20	16.5%	101	83.5%
però	58	89.2%	7	10.8%	14	63.6%	8	36.4%
ma	45	56.3%	35	43.8%	16	55.2%	13	44.8%
quindi	44	57.9%	32	42.1%	17	51.5%	16	48.5%
poi	45	63.4%	26	36.6%	6	37.5%	10	62.5%
se	30	26.5%	83	73.5%	17	36.2%	30	63.8%
così	20	44.4%	25	55.6%	7	35.0%	13	65.0%
che	11	3.5%	302	96.5%	12	8.6%	127	91.4%
*	492	33.7%	966	66.3%	184	33.1%	372	66.9%

Table 9: Distribution of the 10 most frequent connectives (CONN) in training and test sets with the frequencies of their non-discourse connective usages (O).

Data Pre-processing

The discourse annotation of the LUNA corpus was done using text extracted from manual transcriptions that do not contain word beginning and end time information. Thus, in order to be able to extract acoustic features for connective candidates, the text is aligned with the speech signal. The boundaries of words in the speech signal are obtained using forced alignment between word-level manual transcription and the speech signal within the manually segmented turn of a dialog. For the forced alignment, we use Automatic Speech Recognizer (ASR) that was trained on the LUNA Human-Human corpus using Kaldi [40] with Speaker Adaptive Training (SAT). The Word Error Rate (WER) of the ASR on the LUNA Human-Human test set is 39.7%. The ASR system is also used to produce automated transcriptions of manually segmented turns to match with discourse connectives in the corpus. Due to the error rate of the ASR, about 3.7% of discourse connectives are not recognized and are removed from data (see Table 8). After that, forced alignment is used to extract the features discussed in the following subsection.

The MFCC features were spliced, taking 3 frames from each side of the current frame, followed by Linear Discriminant analysis (LDA) and Maximum Likelihood Linear Transform (MLLT) feature-space transformations reducing the feature space. Then, we used Speaker Adaptive Training (SAT). The Word Error Rate (WER) for our ASR system is 39.70% on official LUNA-HH test set.

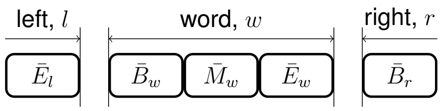
Features

Several sets of features are extracted for supervised machine learning from the force-aligned ASR output. Lexical features are tokens and, since we train connective specific models, they appear only as lexical context features. The difference between manual and ASR output context tokens is negligible (lower that 0.1%). The rest of the features used in the experiments is described below.

Duration and Silence Features: The time it took to utter a connective candidate and the duration of pauses before and after it might also carry information relevant to discourse. Thus, word and silence durations are extracted from the forced alignment and used as features for classification (3 features).

Acoustic Features: Acoustic frame-wise Low-Level Descriptors (LLD) are extracted using openSMILE [17] with the Frame Size = 25 ms and Frame Step = 10 ms. The extracted LLD are *prosodic* features (3) – fundamental frequency ($F0$), pitch, and loudness, all with their derivatives (2 per feature) – and *spectral* features (2) – flux and centroid (11 LLD in total).

We consider 3 segments – connective candidate token (w) and its left (l) and right (r) contexts (up to 2 words taken as a single segment), and acoustic features are extracted for each seg-



$$f_{i,1} = \bar{M}_w - \bar{B}_w;$$
$$f_{i,2} = \bar{E}_w - \bar{M}_w;$$
$$f_{i,3} = \begin{cases} \bar{B}_w - \bar{E}_l & : l \neq \emptyset \\ \bar{B}_w - \bar{m}_{spk} & : \text{otherwise} \end{cases};$$
$$f_{i,4} = \begin{cases} \bar{B}_r - \bar{E}_w & : r \neq \emptyset \\ \bar{m}_{spk} - \bar{E}_w & : \text{otherwise} \end{cases};$$

Figure 3: Acoustic difference feature generation. Intra-word variation is represented by features $f_{i,1}$ and $f_{i,2}$ and cross-word variation by $f_{i,3}$ and $f_{i,4}$; where i is a Low-Level Descriptor (LLD).

ment separately. In each segment, per frame feature values are normalized by Z-score with speaker-based mean ($\bar{m}_{spk}$) and its standard deviation (σ_{spk}), which are calculated using all the dialog turns of a corresponding speaker that do not contain discourse connective candidates. For normalization purposes overlapping turns are considered as a separate speaker.

Each segment S is later split into three parts – beginning (B_S), middle (M_S) and end (E_S); and arithmetic mean of all the frame feature values is calculated for the segment parts: $\bar{B}_S$, $\bar{M}_S$, and $\bar{E}_S$. As a result, there are 9 values per LLD for a connective candidate: means of beginning, middle and end parts for a candidate itself and its right and left contexts. Consequently, each candidate is represented by 99 acoustic features (11 LLD * 9 parts).

Acoustic Difference Features: In order to capture changes in the prosody within a word or with respect to context, using the acoustic features described above, we generate four acoustic difference features. For intra-word variation we calculate two differences – between middle ($\bar{M}_w$) and beginning ($\bar{B}_w$) and between end ($\bar{E}_w$) and middle ($\bar{M}_w$) parts of the word segment. For cross-word variation, the computed differences are between beginning part of the word ($\bar{B}_w$) and the final part of left context ($\bar{E}_l$) and between beginning part of the right context ($\bar{B}_r$) and the end part of the word ($\bar{E}_w$). In case of missing left or right context, the difference is computed with respect to the speaker mean $\bar{m}_{spk}$ (See Figure 3). The difference features are computed for each of 11 LLD; consequently, there are 44 difference features in total (11 LDD * 4 differences).

2.3.3 Experiments and Results

Our goal is to study the relevance of the lexical and acoustic contexts for discourse connective detection from speech. The task is cast as binary discourse connective *vs.* all classification using acoustic and lexical features. Context is defined as features extracted from the segments to the left and right of a connective candidate (i.e. $S \in \{l, r\}$). Since pauses before and after word are not in the word segment, they are considered as context.

For classification we use AdaBoost algorithm [20] implemented in icsiboost [18]. All models are trained on 1,000 iterations, and, despite the unbalanced nature of our data, we do not apply any balancing techniques at this stage.

We describe four sets of experiments: (1) using acoustic features from only connective candidate segment (i.e. *without* context); (2) using acoustic features from only context segments (i.e. *from* context); (3) using acoustic features from all the segments (i.e. *with* context); and (4) using lexical context in isolation and with acoustic features. For settings 1-3 we train and evaluate models on the three sets of features described above – durations, acoustic features, and acoustic difference features – and their combination through vector fusion. For setting 4 we fuse lexical context with the fused vectors from settings 1-3.

Standard precision, recall and F_1 are used as evaluation metrics; however, due to space considerations, we report only F_1. We also compute a micro-averaged F_1 for whole connective set and test it for statistical significance. Statistical significance is measured using McNemar's χ^2 test with Yates' correction.

The Baseline The baseline of discourse connective detection is computed as a majority decision. That is, if a word appears more frequently as a connective in the training set, it is labeled as such in the test set. While some connectives have relatively high baselines (e.g. *perchè*: $F_1 = 90.91$ and *però*: $F_1 = 77.78$), the micro-averaged F_1 is low (53.99) since frequent discourse connectives *e* and *allora* mostly appear in non-discourse roles. For comparison, token only model on PDTB yields $F_1 = 75.33$ [37].

An alternative to training per connective models is a binary classification pooling all connectives together. The majority baseline for models trained using only connective tokens is identical for both settings. In preliminary pooled evaluation, only lexical context features have produced models outperforming the baseline. Thus, we focus on connective specific models and evaluate the relevance of acoustic and lexical context for each connective separately.

Connective Detection *without* Acoustic Context Connective detection *without* context implies not using features outside of the connective candidate time frame; thus, they are word duration, acoustic features extracted from the word segment and within-word acoustic difference features, and their combinations. Results are reported in Table 10. All micro-averaged F_1, except for duration model (**D**), are significantly lower than the baseline. However, we observe that all the features contribute to the detection of a specific connective. Specifically, to the detection of the connectives mostly having non-discourse usages (i.e. *e*, *allora*, *se*, and *così*). Fusion of the features (**ALL**$_N$) does not produce the best model for all, but *se*.

Conn.	BL	D	Ac$_N$	Diff$_N$	ALL$_N$
e	0.00	11.32	25.71	**46.58**	36.62
perchè	90.91	88.00	89.19	84.93	83.33
allora	0.00	0.00	5.71	**17.02**	11.76
però	77.78	77.78	74.29	74.29	68.75
ma	71.11	68.75	60.00	64.52	64.52
quindi	68.00	61.90	51.28	54.05	66.67
poi	54.55	47.06	52.63	46.15	33.33
se	0.00	21.43	37.50	25.00	**38.89**
così	0.00	28.57	**62.50**	40.00	50.00
che	0.00	0.00	0.00	0.00	0.00
Micro	53.99	50.46	49.86	51.37	51.54

Table 10: F_1 of models trained using non-contextual features: duration (**D**), acoustic features (**Ac**$_N$) and intra-word acoustic difference features (**Diff**$_N$) in isolation and in combination (**ALL**$_N$). **BL** is the majority baseline.

Connective Detection *from* Acoustic Context Connective detection *from* context implies using only features extracted from the left and right context of a connective candidate and their fusion. Context also includes lexical tokens in the window ± 2, which are evaluated separately. Micro-averaged F_1 in Table 11, even thought often higher, are not significantly different from the baseline. Neither they are significantly different from the setting without context. Similar to the setting without context, the acoustic context features do contribute to the detection of the connectives mostly having non-discourse usages; however, they also contribute to the detection of others. Fusion of the features (**ALL**$_C$) does not produce the best model for all, but *così*.

Connective Detection *with* Acoustic Context Connective detection *with* context implies that we can use all the features. Micro-averaged F_1 in Table 12 are not significantly different from either baseline or the previous settings, for all but acoustic features: F_1 for all acoustic features (**Ac**) is significantly higher than for the models without context (**Ac**$_C$). Similar to the previous settings, there are individual contributions to specific connectives and the fusion produces the best model only for *così*.

Connective Detection with Lexical Context In this setting we evaluate the relevance of lexical context in isolation and through vector fusion with all speech derived features (durations, acoustic and acoustic difference) in the previous three settings. The lexical context is tokens in the window of ± 1 or ± 2 tokens. Results are reported in Table 13 (for space considerations we report only ± 1 window performance). The micro-averaged F_1 for lexical context in the window of ± 1 tokens performs significantly better than the baseline, while in the window

Conn.	**BL**	**S**	**Ac$_C$**	**Diff$_C$**	**ALL$_C$**
e	0.00	36.92	**56.47**	48.84	43.04
perchè	90.91	89.19	**91.89**	80.00	90.67
allora	0.00	0.00	15.38	**19.35**	13.79
però	77.78	77.78	**87.50**	74.29	77.78
ma	71.11	**71.43**	55.56	62.50	58.82
quindi	68.00	68.09	48.48	**70.97**	64.52
poi	54.55	54.55	47.06	15.38	37.50
se	0.00	0.00	**33.33**	**33.33**	22.22
così	0.00	0.00	42.11	53.33	**58.82**
che	0.00	0.00	0.00	**11.11**	0.00
Micro	53.99	55.49	**56.45**	53.41	55.06

Table 11: F_1 of models trained on only acoustic contextual features: silence durations (**S**), acoustic features from context (**Ac$_C$**), and cross-word acoustic difference (**Diff$_C$**) in isolation and in combination (**ALL$_C$**). **BL** is the majority baseline.

of ± 2 is not. Thus, ± 1 window is used for vector fusion with other features.

The addition of lexical context to the speech-derived features does not produce significant changes to micro-averaged F_1. All acoustic-lexical models are not significantly different from the baseline or their equivalents without lexical context. The lexical context model with ± 1 token window is significantly better than the rest.

However, we again observe that individual connective performances are boosted. For connectives *allora*, *quindi* and *se* the fusion of acoustic features with lexical context produces the best results. In order to estimate the upper bound of the model combination (which could be achieved using features selection on the development set) we calculate the oracle of the best models per connective (**O**). The micro-averaged F_1 of the oracle is 68.53.

Overall, we observe that connectives behave differently with respect to acoustic and lexical context. Half of the connectives (*e*, *perchè*, *ma*, *che*, *così*) achieve the best results using only lexical context. The rest is quite diverse: *se* using lexical, but not acoustic context; *allora* using lexical and acoustic context without the features from the word segment, *quindi* using both lexical and acoustic contexts with the features from the word segment. The remaining two connectives *poi* and *però* perform better without lexical context: *però* using just acoustic features from context and *poi* using all the acoustic features; no acoustic difference or duration features for both.

All these differences lead us to conclude that discourse connectives are not uniform and different features are required to distinguish them from their non-discourse connective usages.

Conn.	BL	DS	Ac	Diff	ALL
e	0.00	33.85	**48.00**	36.62	32.43
perchè	90.91	84.93	89.47	84.06	90.67
allora	0.00	0.00	17.65	**18.75**	7.69
però	77.78	74.29	77.78	76.47	74.29
ma	71.11	76.47	68.57	**77.78**	62.50
quindi	68.00	61.11	60.61	64.52	**70.27**
poi	54.55	47.06	**66.67**	50.00	40.00
se	0.00	10.00	**42.86**	18.18	14.81
così	0.00	28.57	52.63	50.00	**53.33**
che	0.00	0.00	0.00	**13.33**	0.00
Micro	53.99	52.12	**58.95**	53.33	52.87

Table 12: F_1 of models trained on both contextual and non-contextual features: word and silence durations (**DS**), all acoustic features (**Ac**), all acoustic difference features (**Diff**), and their vector fusion (**ALL**). **BL** is the majority baseline.

Conn.	BL	L$_1$	ALL$_N$	ALL$_C$	ALL	*O*
e	0.00	**57.97**	39.47	39.02	37.84	*57.97*
perchè	90.91	**91.89**	87.67	87.67	90.67	*91.89*
allora	0.00	16.67	14.29	**31.25**	7.69	*31.25*
però	77.78	66.67	72.73	77.78	74.29	*87.70*
ma	71.11	**78.05**	55.17	50.00	66.67	*78.05*
quindi	68.00	44.44	66.67	66.67	**74.29**	*74.29*
poi	54.55	**57.14**	33.33	28.57	40.00	*66.67*
se	0.00	40.00	**47.06**	23.08	20.69	*47.06*
così	0.00	**66.67**	50.00	**66.67**	42.86	*66.67*
che	0.00	**86.96**	0.00	0.00	0.00	*86.96*
Micro	53.99	**64.93**	54.29	54.08	54.60	***68.53***

Table 13: F_1 of models trained on using lexical context in the window of ± 1 tokens (**L**$_1$) in isolation and in combination with other features: acoustic features without context (**ALL**$_N$), acoustic features from context (**ALL**$_C$), and all acoustic features (**ALL**). **BL** is the majority baseline and **O** is the oracle of the best performing connective specific models.

2.3.4 Conclusion

We have observed that both lexical and acoustic context have mixed effect on the task of Discourse Connective Detection from speech. While lexical context model significantly outperforms the baseline with $F_1 = 64.93$, the oracle of combination with acoustic context has $F_1 = 68.53$. The conclusion is that the task of discourse connective detection is hard, but lexical features provide enough discriminative power to get improvement of more than 10 points over the majority baseline.

3 Task 4.2: Extracting event and temporal structure from conversations

In this task we develop tools for identifying events and temporal relations in conversations.

3.1 Applications

As a baseline implementation for English, we adapted for SENSEI a combined GATE pipeline originally developed in the ARCOMEM project [33; 34]. In Period 3 (since the work described in D4.2), we have modified it to add temporal extraction, so it now consists of the following GATE processing resources. We removed the condition that skipped documents which TextCat tagged as not in English, because this was not reliable enough for short social media texts and in our main use case (comments in *The Guardian*) we could rely on the fact that they are in English.

1. Basic NLP tasks for English (tokenization, sentence-splitting, POS-tagging, lemmatization).
2. Named entity recognition for English using ANNIE (gazetteers and rules), and orthographic coreferencing of named entities.
3. Noun phrase and verb phrase chunking.
4. Date normalization (this will be especially useful for anchoring temporal expressions in the near future).
5. Event detection using gazetteers (currently oriented towards financial and major political events and industrial action) and rules.
6. Event detection using a large gazetteer of verb nominalizations and rules.
7. Temporal extraction ("TIMEX" detection) using the GATE-Time [15] plugin, which uses the HeidelTime[1] tagger.
8. Sentiment detection using gazetteers and rules.
9. Processing the annotations to select the important ones (events, temporal expressions, opinions) for transfer back to the conversational repository.

[1] `https://github.com/HeidelTime/heideltime`

In Period 3, we also developed an alternative version using the GATE SentiStrength [32] plugin instead of the gazetteers and rules in Step 8 above. The SentiStrength system [53] produces good results but the core `jar` file (required to use the GATE plugin) is available only under a commercial licence except for academic research, so we will publish this component with both GATE applications but without that core file, which users would have to obtain from the University of Wolverhampton[2] under either the commercial or academic licence.

The sentiment detection component used is determined by the GATE application selected in the run-time configuration file, and we provide files for each option.

3.2 Integration

The *gate-pipelines* Java wrapper component, developed specifically to interact with the SENSEI document repository and described in D4.2, has been improved in various ways, most notably to allow repository document features to be copied into the GATE documents before they are processing by the GATE applications. We added this feature principally to allow document (comment) timestamps to be used by the HeidelTime tool described above.

The wrapper and GATE pipeline were successfully used in the “shared task” of linking readers’ comments to sentences in newspaper articles, as reported at SIGDIAL [3] and MultiLing [2].

3.3 Conclusion

We have provided components for event, time, and sentiment detection, as well as for running GATE applications in general, and integrated them successfully with the conversational repository. We were not able to use their outputs in the formal evaluation for social media described in D6.3 and D1.4, but we have developed an additional version of the prototype user interface for *The Guardian*’s comments which colour-codes sentiment in the users’ comments.

[2] `http://sentistrength.wlv.ac.uk/`

4 Task 4.3: Intra-document coreference for conversations and social media

In this task we tune statistical intra-document coreference algorithms to work with conversational and social media data using the BART platform.

4.1 Depositing coreference data in the Repository

In order to deposit coreference data into the Repository which was described in D5.1 of period 1, first a converter from BART's output format (MMAX) to JSON had to be developed. A Data Flow diagram illustrating the full process is shown on Figure 4.

The overall process goes as follows. First, documents are retrieved from the repository in JSON format (each individual comment is considered a separate document). Next, a news article and all its comments are assembled into one composite document and temporarily stored as text with tab indentations to preserve reply structure. Then, the composite text document is processed by BART whose output is MMAX (standoff XML). Next, the MMAX output is converted to JSON and finally posted back to the repository.

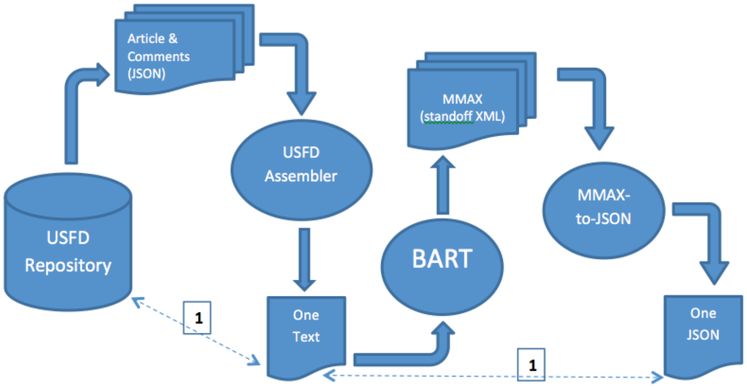


Figure 4: Data Flow Diagram for BART and Repository Integration.

A sample snippet of JSON output is the following:

```
{"BART_Coreference":[
    {"start":0, "end":11, "type":"markable", "chain_ID":100},
    {"start":127, "end":138, "type":"markable", "chain_ID":100},
```

Table 14: USFD Corpus.

Number of words	68853
Number of sentences	3802
Number of markables	22107
Number of coreference chains	2333

```
    {"start":188, "end":190, "type":"markable", "chain_ID":100},
    ...
 ]
}
```

In the above JSON representation, coreference chains are allowed to cross over documents in the repository via a unique global ID, named *chain_ID*. It is important to note the differences; within the composite document representation used by BART coreference chains live within the same document, but when stored in the repository each comment is stored as a separate document and hence the global ID allows for coreference chains to stretch across different documents in the repository.

4.2 Adapting intra-document coreference to social media

4.2.1 The USFD Corpus

In order to annotate the USFD corpus with coreferences we followed the same procedure as for the ONFORUMS corpus [26]; the annotation scheme is a variant of the LiveMemories annotation scheme [45] which in turn is based on the ARRAU annotation scheme [39]. In this corpus all noun phrases are taken as mentions, and the whole noun phrase is considered (with all its embedded NPs). All anaphoric relations of identity between any pairs of mentions are annotated. Coordinations are also treated as mentions, and annotated.

Key corpus statistics are shown in Table 14.

As can be seen from Table 14, this is the largest data set of all three sets that were annotated during the project, consisting of nearly 69K words. It is also a data set prepared and ready for the shared task ONFORUMS17 as explained in D7.5 and D7.6.

4.2.2 The Effects of Layout on Coreference

The effect of discourse structure on anaphora (coreference) resolution was for many years one of the main areas of research on anaphora [14; 19; 21; 22; 38; 43; 59] but has been paid much less attention in recent years. This is due to at least two reasons. First, discourse structure effects are much weaker in news text (the focus of much recent research on coreference) than in task-oriented dialogue (the type of data that had motivated initial work such as Reichman's and Grosz and Sidner's). And second, it is difficult to identify the discourse structure of a text with sufficient accuracy.

The motivation for studying the effects of Layout on Coreference is based on two observations: that discourse structure effects are much stronger in **comment threads** on **online forums**; and that the structure of such threads can be identified much more easily on the basis of the reply structure of the thread. Consider the following example from the comment thread in response to an article in *The Guardian*:

(1) C_1**:** The First Sea Lord, Sir George Zambellas, came closest to expressing it, calling the $3bn ship "a national instrument of power". Who is he planning to invade now?
$\hookrightarrow C_2$**:** Which 'nation'?
$\hookrightarrow C_3$**:** He said it was an example of a big nation demonstrating what they do ... spend countless billions on a vessel that will at best have no aircraft for at least 6-10 years and when there is enough support vessels to defend this hulking lump. Lets gloss over the anti ship ballistic missiles that could render them sitting ducks.
$\longrightarrow C_4$**:** Agree regarding the time scale for fixed wing aircraft, however I'm not so sure about your statement with regards to anti ship missiles...
$\longrightarrow C_5$**:** There's no such thing as an anti-ship ballistic missile...
$\longrightarrow C_6$**:** I doubt that a ballistic missile would render them sitting ducks because they have to be aimed at a fixed point and aircraft carriers are not fixed, they are moving...

In Example (1), comments $C_{[1-6]}$ constitute a typical online conversation thread; comments C_2 and C_3 are replies to comment C_1 (first level) and comments $C_{[4-6]}$ are replies to comment C_3 (second level). A key entity in this conversation thread is *anti ship ballistic missiles* and the context in which the whole thread lives in is a news article titled 'Supercarrier made in Britain hailed as flagship for Better Together campaign'.

Now whilst the news article triggered numerous discussions on war ships, carriers and the Scottish Independence referendum, this was the only conversation thread which discussed *anti ship ballistic missiles*, and hence, this entity is not accessible from outside this thread. And, crucially, the structure of this thread can be readily recovered from the metadata about the comment thread provided by *The Guardian*'s website.

To experiment with anaphora resolution in the online forums domain, we rely on the publicly available BART toolkit [57; 58]. BART is a modular framework that allows for fast development of coreference models. In particular, BART supports very straightforward feature engineering, making it an ideal system for integrating different types of knowledge into coreference resolvers and evaluating their impact.

We have further extended BART by defining a number of new features capturing and exploiting the structure of online conversation threads (see Example (1)). We define features around the notion of 'accessibility' (in the discourse sense), which indicates whether a potential antecedent for an anaphor is accessible or not to the anaphor depending on its position in the thread. Currently we are working with four boolean features, two modeling strict accessibility and two on loose accessibility:

1. *Direct Strict Accessibility*: on iff antecedent is within the current post or the directly replied-to post
2. *Transitive Strict Accessibility*: on iff antecedent is on the path of the reply/layout structure, allowing for transitive replies
3. *Direct Loose Accessibility*: on iff antecedent is anywhere within the same conversation branch (no transitive replies)
4. *Transitive Loose Accessibility*: on iff antecedent is anywhere within the same conversation thread (including transitive replies)

We experimented with three machine learning models: Maximum Entropy (MXE), Decision Trees (C4.5) and Support Vector Machines (SVM). And we tried five different feature configurations: no thread features (baseline), only direct, only transitive, direct and transitive and all features.

We report here (see Table 15) results on 8 fold cross validation across five different CoNLL metrics: MUC, B^3, CEAF$_e$, CEAF$_m$ and MELA[3] produced by the official CoNLL scorer [41].

We set a statistical hypothesis test where the null hypothesis (H_0) we are trying to reject is that there is no difference between the four feature configurations involving thread features and the baseline. In all cases we train the baseline along with the other feature configurations to measure only the thread features contribution.

Then, within each model all feature configurations are tested against the baseline and we have significant improvements as follows (see scores in bold face in Table 15):

- SVM on B^3 metric with all features,
- Decision Trees (C4.5) on CEAF$_e$ metric with direct feature only, and
- Decision Trees (C4.5) on CEAF$_m$ metric with direct feature only

[3] This last one is a composite metric given by the following formula: $MELA = (MUC + B^3 + CEAF_e) * 0.33$

		MXE			C4.5			SVM		
		R	P	F_1	R	P	F_1	R	P	F_1
MUC	baseline	43.36	50.36	46.49	42.54	50.85	46.23	41.79	49.48	45.19
	direct only	43.63	49.97	46.4	42.39	51.63	46.4	42.1	49.61	45.38
	transitive only	43.43	50.38	46.56	42.66	51.04	46.36	42.02	49.79	45.44
	direct and transitive	43.12	51.08	46.58	42.93	51.2	46.54	41.73	49.82	45.29
	all features	43.29	51.78	46.97	43.07	51.65	46.8	41.64	50.63	45.55
B^3	baseline	44.86	57.38	50.23	44.54	58.41	50.47	44.36	57.4	49.97
	direct only	44.86	56.56	49.99	44.56	59.04	50.72	44.42	57.3	49.99
	transitive only	44.90	57.23	50.23	44.6	58.27	50.47	44.38	57.55	50.05
	direct and transitive	44.67	57.86	50.37	44.62	58.26	50.47	44.34	57.79	50.11
	all features	44.76	58.64	50.72	44.69	58.79	50.74	44.32	58.6	**50.41**†
$CEAF_e$	baseline	42.83	49.42	45.86	43.33	49.26	46.08	42.74	48.87	45.57
	direct only	42.39	49.48	45.62	43.73	49.31	**46.32**†	42.76	49.05	45.66
	transitive only	42.86	49.55	45.94	43.32	49.33	46.11	42.83	48.96	45.66
	direct and transitive	43.11	49.41	46.01	43.27	49.39	46.1	43.0	48.95	45.75
	all features	43.4	49.42	46.17	43.31	49.24	46.04	43.25	48.8	45.83
$CEAF_m$	baseline	44.1	51.03	47.31	44.24	51.21	47.47	43.48	50.33	46.65
	direct only	43.89	50.81	47.1	44.52	51.54	**47.77**†	43.54	50.41	46.72
	transitive only	44.16	51.12	47.38	44.16	51.12	47.38	43.54	50.40	46.72
	direct and transitive	44.11	51.07	47.34	44.17	51.13	47.39	43.62	50.49	46.8
	all features	44.6	51.62	47.85	44.44	51.44	47.68	43.79	50.69	46.99
MELA	baseline	-	-	47.05	-	-	47.12	-	-	46.44
	direct only	-	-	46.86	-	-	47.34	-	-	46.54
	transitive only	-	-	47.1	-	-	47.17	-	-	46.57
	direct and transitive	-	-	47.18	-	-	47.23	-	-	46.58
	all features	-	-	47.47	-	-	47.38	-	-	46.79

Table 15: Coreference Resolution Performance, 8 X Validation († sig. at $\alpha = 0.05$).

This is three cases out of 60 test cases (4 x 5 x 3 , i.e., 4 different feature configurations, 5 metrics, 3 models). The MELA scores for SVMs and Decision Trees are near significance levels (e.g., $t = 1.78$, critical $t = 1.89$). Sign Test on 8 folds is difficult, for example, we have cases with 7 pluses and 1 minus which is $p = 0.07$ (again, almost 0.05), thus with 10 folds would be better.

On the other hand we have not looked at differences across models. For example, in absolute terms the Maximum Entropy models (MXE) seem to produce the best scores across metrics.

4.2.3 The Effects of Coreference on News Comment Clustering

We also ran experiments on a higher level task in order to provide a task-based scenario for evaluating Coreference improvements. The task at hand is news comments clustering.

To this end, we have adopted the graph based approach described by Aker et al. [4]. Nodes in the graph represent the comments and edges are created with weights capturing the similarity between the comments. The resulting graph is converted to a square matrix whose rows and columns correspond to nodes in the graph and whose cell values to weights. Two operations are run on the square matrix to obtain the clusters: expansion and inflation. The expansion operator is responsible for allowing flow to connect different regions of the graph. The inflation operator is responsible for both strengthening and weakening this flow.

To generate a weight between two nodes or comments, Aker et al. [4] applied various features ranging from similarity measures such as cosine similarity over the comment words and named entity overlaps to features such as same thread and reply relationship between the comments. We have extended their list of features with anophoric relationships to investigate the impact of such pieces of information in comment clustering. The following features are added in addition to the ones described by Aker et al.:

- *Has same annotation id*: This is a binary feature and returns 1 if both comments share a common annotation id otherwise 0.
- *Cosine between the annotation IDs*: We determine all the annotation' ids from both comments and compute the cosine angle between annotation ids. Beforehand a dictionary vector is created consisting of all the unique annotation ids, e.g. “1,2,3,4,5”. Each comment is then represented using this vector. For each annotation id either a “1” or “0” is included in the vector depending whether that id appears in the comment or not. E.g. if the comment contains annotation ids “2” and “4” its vector gets the space “0,1,0,1,0”. Based on these vectors we determine the cosine angle between two comments.
- *Cosine between the mentions in the anaphora chain*: A chain in the anaphora resolution output consists of an antecedent and all anaphors referring to it. For each anaphor in each comment we determine its anaphora chain, collect the words in the antecedent (e.g. “Buckingham Palace”) and as well as of the anaphors (e.g. ”The Palace”). Then we merge the words of the anaphora chains in word vectors and compute the cosine angle between the merged versions of the word vectors. In the cosine vectors we use single words.
- *Cosine between the contents enriched with the mentions in the anaphora chain*: In the previous feature we used only the anaphora chains. In this feature we use the words in the comments and the anaphora chains to compute the cosine score.
- *Word2Vec on mentions in the anaphora chain*: Word embeddings using Word2Vec [35] have been extensively used to measure the semantic similarity between words. Our word embeddings comprise the vectors published by Baroni et al. [7]. Using Word2Vec we compute the cosine angle between the mentions in the anaphora chain. Before computing the cosine angle we first remove from each comment stop-words as well as punctuation, query for each word its vector representation and create a averaged sum of the

word vectors. The number of remaining words in each comment is used to average that comment. Finally, we use the resulting averaged sum vectors and determine their similarity using the cosine similarity measure.

- *Word2Vec on mentions in the anaphora chain as well as the comment contents*: In addition to the previous feature we also add the comment contents and compute the cosine between the Word2Vec representations.

Following Aker et al. [4], we sum the features using weighted linear combination and train the weights using linear regression[4] and gold standard clusters.

We run the experiments on clustering matching the cross validation experiments from the previous section[5] and, hence, we use the same feature combinations to train models and produce coreference outputs corresponding to each different feature combination. Then, we use the various coreference outputs on a more general task of news comment clustering and gauge performance changes. That is, the different coreference outputs are used to extract anaphoric features described above in the first part of this Section 4.2.3, then these features are added to the feature set reported by Aker et al. [4], comment clusters are generated and evaluation results are computed. We use as baseline the setting reported by Aker et al. without any anaphoric features. The evaluation results are computed using fuzzy BCubed Precision, Recall and F-Measure metrics reported in [25] and [23] and the gold standard clusters. According to the analysis of formal constraints that a cluster evaluation metric needs to fulfill [5], fuzzy BCubed metrics are superior to other know metrics such as Purity, Inverse Purity, Mutual Information, Rand Index, etc. as they fulfill all the formal cluster constraints: *cluster homogeneity*, *completeness*, *rag bag* and *clusters size versus quantity*. The results are shown in Table 16.

	R	**P**	**F1**
No Coreference	44.8%	42.5%	43.0%
Gold Coreference	45.0%	42.5%	44.0%
BART 2.0	44.5%	42.8%	43.0%
+ all access. features	45.5%	42.3%	44.0%
+ direct strict	45.5%	42.3%	44.0%
+ transitive strict	44.0%	42.3%	43.0%
+ dir. & transit. strict	44.0%	42.3%	43.0%

Table 16: Impact of Coreference on news comment clustering.

The top two rows in Table 16 are the baseline which is topic clustering without using coreference, and the upper bound, the case when gold coreferences are used from the human

[4] We used Weka (http://www.cs.waikato.ac.nz/ml/weka/) implementation of linear regression.

[5] Please note that at the time of writing of this manuscript, we only had results from a cross validation on four folds, that is, on half of the data set from the previous section.

annotation.

From Table 16 we can see that there is some improvement by using coreference in news comment clustering, in particular, we improve from $F1 = 43.0$ in the baseline case of not using coreference to $F1 = 44.0$ by using all accessibility features, though we note the improvement is not statistically significant. However, we do not get significance by using the human-annotated gold coreferences either, which suggests that more work needs to be done on how coreference is harnessed to improve clustering. Moreover, we note that the way we define our anaphoric features inherently leads to high redundancy and correlations amongst features, and that our method of choice for combining them, linear regression, may be sensitive to correlations between features, and hence, might not be ideal. All this to be explored in future work.

5 Task 4.4: Inter-document coreference for conversations and social media

5.1 Experiments on inter-document coreference in the social media domain

Similarly as in Period 2, for our inter-document coreference experiments in Period 3 we again used the JRC-Names resource developed at the Joint Research Centre of the European Commission [49]. It is a highly multilingual named entity resource (persons and organisations), which consists of large lists of names and their multiple (in the order of hundreds) spelling variants and transliterations across scripts (Latin, Greek, Arabic, Cyrillic, Japanese, Chinese, etc.). Plugged into a standard pattern matcher, it can be used for multilingual named entity disambiguation across documents. Since it was created by analysing millions of news articles in many languages and over many years[6], it can be expected that its optimal performance would be in the news domain.

We ran JRC-Names over the text versions of the USFD corpus files for English (see previous section for corpus statistics). The number of different entities found per file for English and Italian is summarised in Table 17.

Table 17: Number of entities found per file using the JRC entity disambiguator.

File	Entities	File	Entities
articleCommentBartInput0	5	articleCommentBartInput5	23
articleCommentBartInput1	9	articleCommentBartInput9	5
articleCommentBartInput2	21	articleCommentBartInput10	11
articleCommentBartInput3	7	articleCommentBartInput11	12
articleCommentBartInput4	26	articleCommentBartInput12	16

The full output from JRC-Names after processing the file from which the ‘ballistic missile’ example was drawn (see previous section), that is, file `articarticleCommentBartInput4`, is the following:

```
found entity id = 342 type p as Menzies Campbell (862)
found entity id = 227606 type p as Danny Alexander (991)
found entity id = 507190 type o as BAE SYSTEMS (7510)
found entity id = 507190 type o as BAE Systems (34602)
```

[6] See `http://emm.newsexplorer.eu/`

```
found entity id = 13388 type o as BAE SYSTEMS (7510)
found entity id = 13388 type o as BAE Systems (34602)
found entity id = 118579 type p as Lord Robertson (972)
found entity id = 19266 type p as David Cameron (654)
found entity id = 19267 type p as George Osborne (788)
found entity id = 52043 type p as Ed Miliband (818)
found entity id = 36 type p as Gordon Brown (841)
found entity id = 109848 type p as William Wilberforce (31439)
found entity id = 12637 type p as Phillip Hammond (925)
found entity id = 9138 type o as Red Cross (23811)
found entity id = 87261 type o as Monty Python (15020,18052)
found entity id = 338051 type o as House of Lords (13146)
found entity id = 2170 type p as Alex Salmond (692)
found entity id = 10101 type o as EU (5707,13275,13935,17088)
found entity id = 80780 type o as Cold War (29634)
found entity id = 145579 type p as Alan Shepard (11597)
found entity id = 86156 type p as Queen Elizabeth (94,238,544,3536,4332)
found entity id = 1628241 type o as White Paper (35920)
found entity id = 23557 type o as Royal Navy (412,1212,2402,3103,3333,3755,...)
found entity id = 237270 type o as Ministry of Defence (3002)
found entity id = 197075 type o as Google (13138)
found entity id = 456610 type o as Eurofighter Typhoon (5669)
```

Once again, in order to have a sense of coverage of the tool we counted the number of coreference chains in the gold standard annotation. We include these statistics in Table 18. The reason why we show them in a separate table is because they are not directly comparable with the number of entities shown in Table 17, they are only indicative of coverage.[7]

From Tables 18 and 17 we can see that JRC-Names is able to identify roughly between 2% and 10% of the entities represented by the annotated coreference chains. It is worth noting that many of the coreference chains do not represent persons or organisations which are the scope of the JRC-Names resource.

In some ways the JRC-Names output complements that of BART – it provides the bridge of coreference chains across documents, because the JRC-Names disambiguates entities to a unique global id regardless of input. More experiments and an integration at the software level of both tools could provide more insight into the interrelationships.

[7] The set of identified entities is not necessarily subsumed by the set of coreference chains, that is, there may be potentially only partial overlap between the two sets.

Table 18: Number of coreference chains in the gold standard annotation of the USFD corpus.

English	
File	Coreference Chains
articleCommentBartInput0	260
articleCommentBartInput1	294
articleCommentBartInput2	228
articleCommentBartInput3	176
articleCommentBartInput4	269
articleCommentBartInput5	251
articleCommentBartInput9	190
articleCommentBartInput10	159
articleCommentBartInput11	220
articleCommentBartInput12	286

6 Task 4.5: The argumentation structure of conversations

In this section we describe work during the third year of the project on Task 4.5 of Work Package 4 (WP4).

6.1 Automatic Labeling of Argument Structures

On-line social conversations concur to the formation of opinions and shared knowledge which influence decision makers [6]. A large amount of lightly moderated multiparty conversations take place on-line every day in social forums and news blogs [46], and bloggers who participate to these conversations usually express agreement and disagreement with respect to positions and statements, generating a large amount of conversational data.
From a communication analysis perspective, online multiparty conversations are asynchronous and more complex than synchronous conversations, such as dyadic spoken dialogues. In asynchronous conversations, bloggers can reply to any other with text messages or pre-coded actions (e.g. *like* buttons). Despite the labelling of Argument Structures in text is traditionally associated to a well-known task such as Semantic role Labelling, we found more interesting and innovative to extract Agreement/Disagreement Relations (henceforth ADRs). To this purpose we trained a cross-language classifier from asynchronous on-line debates, exploiting the Italian dataset (CorEA) that we collected and annotated during periods 1 and 2 of the SENSEI project (see D2.1 and D4.2). This classifier has been used for template-based summarisation (see D5.3) as well as for monitoring and predicting the outcome of the Brexit campaigning (see D6.3).
In this section we describe the prediction and evaluation of the Agreement/Disagreement argument extraction system.

6.1.1 Prediction of Agreement/Disagreement

Previous work on the prediction of ADRs in asynchronous conversations used different approaches: from the analysis of contextual, dependency and word-based features in the IAC corpus [1] to the exploitation of the overall position of a blogger in threaded discussion forums using data manually annotated with lexical features, emotions, duration and sentiment [62]. In [60] they addressed the problem of ADRs classification on the IAC and AAWD corpora. They exploited many features, including n-grams, Part-Of-Speech tags, sentiment, TF-IDF and used Conditional Random Fields as learning algorithm. They achieved performances ranging from 0.56 and 0.58 on Wiki discussions to 0.74 and 0.67 on political debates and found that in the

AAWD corpus agreement is easier to classify with respect to disagreement, while the contrary is true in the IAC corpus.

We started from the data collected previously in the SENSEI project: The CorEA corpus [12]. To summarize, CorEA is a collection of asynchronous news blogs conversations in Italian. The on-line conversations amongst bloggers, in this case, are originated by an on-line news article and developed within the newspaper social blogging platform. CorEA includes user-annotated mood metadata and manually annotated ADRs at the message level. The reported inter-annotator reliability is k=0.58 on 3 classes ("agree", "disagree", "not applicable") and k=0.87 on 2 classes ("agree", "disagree"). Starting from this dataset, we combined different levels of analysis for feature extraction: blogger level, message level and at the level of relation between messages, in order to investigate what are the best predictors of ADRs in asynchronous on-line conversations in the news media domain.

Features For classification experiments on ADRs, we exploited the features already present in the data and enriched them with new features at the level of messages, bloggers and parent-child relations (relational features). Table 19 reports an overview of the features we used at these levels and in the following subsections we describe each feature class in more detail.

feature	type	count
stylometric features	message	97
topic-word ratio	message	1
sentiment polarity	message	1
discourse relation sense counts	message	4
discourse relation sense ratios	message	4
likes	message	1
replies count	message	1
pair similarity	relational	1
article is parent	relational	1
sentiment match	relational	1
topic match	relational	1
personality types	blogger	5
mood priors	blogger	5
avg replies per message	blogger	1
avg likes per message	blogger	1
topics-message ratio	blogger	1
discourse relation sense sums	blogger	4
discourse relation sense ratios	blogger	4
stance	blogger	1

Table 19: Features used in the experiments at message and blogger levels.

Message-level Features (107) *Discourse Features* (8) are frequency counts and ratios (% from total) of the four top-level relation senses from Penn Discourse Treebank (PDTB) [42]: Comparison, Contingency, Expansion, and Temporal. They are extracted for explicit discourse relations (signaled by connectives such as *but*, *however*, *when*, etc.) using lexical context classifier of [44]; and a connective sense classifier trained and tested on Italian LUNA Corpus [16] annotated with PDTB relations [54] The accuracy of the automatically annotated discourse features is 80%. In addition to this, we included the count of message likes and replies (2) as message features.

Sentiment Polarity Features (2) are text-length normalized sums of the polarized words extracted using OpeNER lexicon[8], and their discretisation into positive, neutral, and negative classes. We evaluated the sentiment annotation on TwITA, a manually annotated dataset of tweets for sentiment analysis [8], accuracy is 66%.

Stylometric Features (97) are basic text statistics (4) such as word count, vocabulary size, average word length; frequency-based features (2) such as frequency of hapax legomena; measures of lexical richness (16) based on word count, vocabulary size, and word-frequency spectrum such as mean word frequency, type-token ratio, entropy, Guiraud's R, Honore's H, etc. [56]; and word length ratios (30) for 1-30 character long words. The feature set also includes character-based ratios (45) for character classes (e.g. punctuation, white space, etc.) and individual characters (e.g. '!', 'a', etc.).

Relation-level Features (4) Relation-level features are extracted considering both child and parent messages (or the article). They include word2vec [36] cosine similarity between parent and child, boolean feature to indicate whether a parent is an article or another message, and two boolean features for matches and mismatches between topics and sentiment polarities expressed in two messages.

Blogger-level Features (22) Blogger-level features are the personality types (5), self-assessed bloggers' mood priors (5); the aggregation (sums and averages) of the message-level discourse (8) features; blogger's stance (1) and the bloggers' topic per message ratio (1). Personality types are defined by the Five Factor Model [13]: extroversion, emotional stability/neuroticism, agreeableness, conscientiousness, openness to experience. These features have been automatically predicted exploiting linguistic cues from the collection of all messages of the same blogger [31]. Accuracy of the prediction, evaluated on an Italian Facebook dataset is 65% [11]. Mood priors encoded in CorEA are: indignation, disappointment, worry, amusement and satisfaction. Stance is defined as the sum of the polarity of messages of a blogger.

[8] `http://www.opener-project.eu/`

6.1.2 Evaluation of Agreement/Disagreement labeling

We addressed the prediction of ADRs at message level as a binary classification task. We discarded all the "NA" labels, balanced the classes and split the data into 66% training and 33% testing. In order to prevent overlappings between the training and test set, we sorted messages by bloggers alphabetically. We experimented with different setting, in order to compare the contribution of different feature types to the classification task. We used a Support Vector Machine as learning algorithm and F1-measure as evaluation metric (see Table 20).

settings	agree	disagree	avg
majority baseline	0.500	0.500	0.500
bag of word baseline	0.550	0.624	0.590
message	0.555	0.554	0.550
blogger	0.634	0.568	0.601
relational	**0.726**	**0.684**	**0.705**
message+blogger	0.618	0.560	0.589
message+relational	0.711	0.675	0.693
blogger+relational	0.726	0.684	0.705
all	0.659	0.629	0.644

Table 20: Result of the classification of ADRs using different combination of message features (mf), blogger features (bf) and relational features (rf), 66% training, 33% test split and a Support Vector Machine as classifier.

6.2 Towards the next edition of the shared task ONFORUMS

As pointed out in D7.5 and D7.6 of the project SENSEI, preparations for a second edition of the shared task ONFORUMS mainly revolved around the preparation and annotation of a new data set, which we have named internally the USFD corpus. The data set was described in detail in Section 4.2.1 of this deliverable and the current state of play is that it is ready for another round of crowdsourcing evaluation on the platform CrowdFlower in the same manner as the evaluation of ONFORUMS 2015. It features the added value of Coreference annotation over a subset of it (over 50%) which can be employed either as a new aspect of the shared task exploring the relationship between agreement structure and coreference or simply to further research in Coreference in the domain of online forums.

7 Conclusion

The end-to-end discourse parsing performances are too low to be directly used for downstream tasks. Dialog act segmentation and classification performances, on the other hand, are acceptable high. Thus, dialog acts were used to filter out utterance segments for the extractive summarization for extrinsic evaluation.

We improved the combined event, temporal expression, and sentiment detection tools and developed an additional version of the prototype user interface which incorporates its results into the output, although this was not ready at the time of the formal evaluation described in D1.4.

Working under the assumption that discourse structure effects on Coreference are stronger in comment threads on online forums than in the standard news domain and that such structure can be identified reliably on the basis of the reply structure of the thread, we devised machine learning features to model thread structure and ran experiments with BART. Using such features we attained significant improvements with certain configurations over a baseline without thread awareness which proves our approach promising. However, work on creating more data and more features is required in order to be able to train models which consistently outperform the baseline.

The second conclusion from the work on Coreference on comment threads is that its effects on higher level tasks, such as Clustering, suggest possibilities for improvement, but a lot of effort needs to be put into how it is harnessed into the task.

On inter-document coreference, we used again the JRC-Names resource developed at the JRC to run experiments on the USFD corpus. We found that JRC-Names is able to identify roughly between 2% and 10% of the entities represented by the annotated coreference chains. This was a lower coverage than the experiments on the shared task dataset from Period 2, possibly due to the new dataset featuring less person and organisation references, and hence, outside of the scope of the JRC tool.

Agreement/disagreement structures proved to be a valuable source of information, as we successfully used it for the prediction of Brexit as well as in the summarisation tasks (see D6.3).

Finally, given the successful completion of the shared task ONFORUMS in 2015 and having now a new data set prepared and ready, we hope to be in a position to run it again in 2017.

Bibliography

[1] Rob Abbott, Marilyn Walker, Pranav Anand, Jean E Fox Tree, Robeson Bowmani, and Joseph King. How can you say such things?!?: Recognizing disagreement in informal political argument. In *Proceedings of the Workshop on Languages in Social Media*, pages 2–11. Association for Computational Linguistics, 2011.

[2] Ahmet Aker, Fabio Celli, Adam Funk, Emina Kurtic, Mark Hepple, and Rob Gaizauskas. Sheffield-Trento System for Sentiment and Argument Structure Enhanced Comment-to-Article Linking in the Online News Domain. `http://multiling.iit.demokritos.gr/file/download/1577`, 2015. [Online; accessed 06-August-2015].

[3] Ahmet Aker, Emina Kurtic, Mark Hepple, Rob Gaizauskas, and Giuseppe Di Fabbrizio. Comment-to-article linking in the online news domain. In *Proceedings of SIGDIAL*, pages 245—-249, Prague, Czech Republic, 2015.

[4] Ahmet Aker, Emina Kurtic, AR Balamurali, Monica Paramita, Emma Barker, Mark Hepple, and Rob Gaizauskas. A graph-based approach to topic clustering for online comments to news. In *Advances in Information Retrieval*, pages 15–29. Springer, 2016.

[5] Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information retrieval*, 12 (4):461–486, 2009.

[6] Matthew Barnidge. The role of news in promoting political disagreement on social media. *Computers in Human Behavior*, 52:211–218, 2015.

[7] Marco Baroni, Georgiana Dinu, and Germán Kruszewski. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *ACL (1)*, pages 238–247, 2014.

[8] Valerio Basile and Malvina Nissim. Sentiment analysis on italian tweets. *WASSA 2013*, page 100, 2013.

[9] Frederic Bechet, Alexis Nasr, and Benoit Favre. Adapting dependency parsing to spontaneous speech for open domain spoken language understanding. In *Interspeech*, 2014.

[10] Sabine Buchholz. chunklink.pl. `http://ilk.uvt.nl/software/`, 2000.

[11] Fabio Celli and Luca Polonio. Relationships between personality and interactions in facebook. In *Social Networking: Recent Trends, Emerging Issues and Future Outlook*, pages 41–54. Nova Science Publishers, Inc, 2013.

[12] Fabio Celli, Giuseppe Riccardi, and Arindam Ghosh. Corea: Italian news corpus with emotions and agreement. In *Proceedings of CLIC-it 2014*, pages 98–102, 2014.

[13] Paul T Costa and Robert R McCrae. The revised neo personality inventory (neo-pi-r). *The SAGE handbook of personality theory and assessment*, 2:179–198, 2008.

[14] D. Cristea, N. Ide, and L. Romary. Veins theory: A model of global discourse cohesion and coherence. In *Proc. of COLING*, pages 281–285, Montreal, August 1998.

[15] L. Derczynski, J. Strötgen, D. Maynard, M. A. Greenwood, and M. Jung. GATE-Time: Extraction of temporal expressions and events. In *Proceedings of the 10th Conference on International Language Resources and Evaluation (LREC)*, 2016. URL `http://www.derczynski.com/sheffield/papers/gate-time.pdf`.

[16] Marco Dinarelli, Silvia Quarteroni, Sara Tonelli, Alessandro Moschitti, and Giuseppe Riccardi. Annotating spoken dialogs: from speech segments to dialog acts and frame semantics. In *Proceedings of EACL Workshop on the Semantic Representation of Spoken Language*, Athens, Greece, 2009.

[17] Florian Eyben, Martin Wöllmer, and Björn Schuller. Opensmile: The munich versatile and fast open-source audio feature extractor. In *Proceedings of the International Conference on Multimedia*, pages 1459–1462, New York, NY, USA, 2010. ACM.

[18] Benoit Favre, Dilek Hakkani-Tür, and Sebastien Cuendet. Icsiboost. `https://github.com/benob/icsiboost/`, 2007.

[19] B. A. Fox. *Discourse Structure and Anaphora*. Cambridge University Press, Cambridge, UK, 1987.

[20] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1): 119–139, August 1997.

[21] B. J. Grosz. *The Representation and Use of Focus in Dialogue Understanding*. PhD thesis, Stanford University, 1977.

[22] B. J. Grosz and C. L. Sidner. Attention, intention, and the structure of discourse. *Computational Linguistics*, 12(3):175–204, 1986.

[23] Eyke Hüllermeier, Maria Rifqi, Sascha Henzgen, and Robin Senge. Comparing fuzzy partitions: A generalization of the rand index and related measures. *Fuzzy Systems, IEEE Transactions on*, 20(3):546–556, 2012.

[24] Richard Johansson and Alessandro Moschitti. Syntactic and semantic structure for opinion expression detection. In *CoNLL 2010*, 2010.

[25] David Jurgens and Ioannis Klapaftis. Semeval-2013 task 13: Word sense induction for graded and non-graded senses. In *Second joint conference on lexical and computational semantics (* SEM)*, volume 2, pages 290–299, 2013.

[26] Mijail Kabadjov, Udo Kruschwitz, Massimo Poesio, Josef Steinberger, Marc Poch, and Hugo Zaragoza. The OnForumS corpus from the Shared Task on Online Forum Summarisation at MultiLing 2015. In *Proceedings of LREC*, Portoroz, Slovenia, 2016.

[27] Karin Kipper, Anna Korhonen, Neville Ryant, and Martha Palmer. A large-scale classification of english verbs. *Language Resources and Evaluation Journal*, 42(1):21–40, 2008.

[28] Taku Kudo. CRF++. `http://taku910.github.io/crfpp/`, 2013.

[29] John Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of 18th International Conference on Machine Learning*, pages 282–289, 2001.

[30] Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. A PDTB-styled end-to-end discourse parser. *Natural Language Engineering*, 20:151 – 184, 2014.

[31] François Mairesse, Marilyn A Walker, Matthias R Mehl, and Roger K Moore. Using linguistic cues for the automatic recognition of personality in conversation and text. *Journal of Artificial Intelligence Research*, 30(1):457–500, 2007.

[32] D.G. Maynard and K. Bontcheva. Challenges of evaluating sentiment analysis tools on social media. In *Proceedings of the 10th Conference on International Language Resources and Evaluation (LREC)*, 2016. URL `http://eprints.whiterose.ac.uk/98769/`.

[33] Diana Maynard and Adam Funk. Automatic detection of political opinions in tweets. In Raul Garcia-Castro, Fensel Dieter, and Antoniou Grigoris, editors, *The Semantic Web: ESWC 2011 Workshops*, pages 88–99. Springer, 2012.

[34] Diana Maynard, Gerhard Gossen, Adam Funk, and Marco Fisichella. Should I care about your opinion? detection of opinion interestingness and dynamics in social media. *Future Internet*, 6(3):457–481, 2014.

[35] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[36] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751, 2013.

[37] Emily Pitler and Ani Nenkova. Using syntax to disambiguate explicit discourse connectives in text. In *Proceedings of the ACL-IJCNLP Conference*, pages 13–16, 2009.

[38] M. Poesio, A. Patel, and B. Di Eugenio. Discourse structure and anaphora in tutorial dialogues: an empirical analysis of two theories of the global focus. *Research in Language and Computation*, 4:229–257, 2006. Special Issue on Generation and Dialogue.

[39] Massimo Poesio and Ron Artstein. Anaphoric annotation in the arrau corpus. In *Proceedings of LREC*, Marrakesh, Morocco, 2008.

[40] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. The kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE, 2011.

[41] Sameer Pradhan, Xiaoqiang Luo, Marta Recasens, Eduard H Hovy, Vincent Ng, and Michael Strube. Scoring coreference partitions of predicted mentions: A reference implementation. In *ACL (2)*, pages 30–35, 2014.

[42] Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. The Penn Discourse Treebank 2.0. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*, 2008.

[43] R. Reichman. *Getting Computers to Talk Like You and Me*. The MIT Press, Cambridge, MA, 1985.

[44] Giuseppe Riccardi, Evgeny A. Stepanov, and Shammur Absar Chowdhury. Discourse connective detection in spoken conversations. In *Proceedings of ICASSP*, Shanghai, China, March 2016. IEEE.

[45] K. Rodriguez, F. Delogu, Y. Versley, E. W. Stemle, and Massimo Poesio. Anaphoric annotation of wikipedia and blogs in the live memories corpus. In *Proceedings of LREC*, Floriana, Malta, 2010.

[46] Carlos Ruiz, David Domingo, Josep Lluís Micó, Javier Díaz-Noci, Pere Masip, and Koldo Meso. Public sphere 2.0? the democratic qualities of citizen debates in online newspapers. *The International Journal of Press/Politics*, pages 1–25, 2011.

[47] Deborah Schiffrin. *Discourse Markers*. Cambridge University Press, 1987.

[48] Helmut Schmid. Improvements in part-of-speech tagging with an application to german. In *Proceedings of the ACL SIGDAT-Workshop*, Dublin, Ireland, 1995.

[49] Ralf Steinberger, Bruno Pouliquen, Mijail Kabadjov, Jenya Belyaeva, and Erik van der Goot. JRC-Names: A freely available, highly multilingual named entity resource. In *Proceedings of RANLP*, Hissar, Bulgaria, 2011.

[50] Evgeny A. Stepanov and Giuseppe Riccardi. Comparative evaluation of argument extraction algorithms in discourse relation parsing. In *The 13th International Conference on Parsing Technologies (IWPT 2013)*, pages 36–44, Nara, Japan, November 2013.

[51] Evgeny A. Stepanov and Giuseppe Riccardi. Towards cross-domain PDTB-style discourse parsing. In *EACL Workshops - The Fifth International Workshop on Health Text Mining and Information Analysis (Louhi 2014)*, pages 30–37, Gothenburg, Sweden, April 2014. ACL.

[52] Evgeny A. Stepanov, Giuseppe Riccardi, and Ali Orkan Bayer. The UniTN discourse parser in CoNLL 2015 shared task: Token-level sequence labeling with argument-specific models. In *The SIGNLL Conference on Computational Natural Language Learning (CoNLL)- Shared Task*, pages 25–31, Beijing, China, July 2015. ACL.

[53] M. Thelwall, K. Buckley, and G. Paltoglou. Sentiment strength detection for the social web. *Journal of the American Society for Information Science and Technology*, 63(1):163–173, 2012.

[54] Sara Tonelli, Giuseppe Riccardi, Rashmi Prasad, and Aravind K. Joshi. Annotation of discourse relations for conversational spoken dialogs. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, 2010.

[55] Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: A simple and general method for semisupervised learning. In *In ACL*, pages 384–394, 2010.

[56] Fiona J. Tweedie and R. Harald Baayen. How variable may a constant be? measures of lexical richness in perspective. *Computers and the Humanities*, 32(5):323–352, 1998.

[57] Olga Uryupina, Alessandro Moschitti, and Massimo Poesio. BART goes multilingual: The UniTN / Essex submission to the CoNLL-2012 Shared Task. In *Proceedings of the Sixteenth Conference on Computational Natural Language Learning (CoNLL'12)*, 2012.

[58] Yannick Versley, Simone Paolo Ponzetto, Massimo Poesio, Vladimir Eidelman, Alan Jern, Jason Smith, Xiaofeng Yang, and Alessandro Moschitti. BART: a modular toolkit for coreference resolution. In *Proceedings of the 2008 Conference of the Association for Computational Linguistics*, pages 9–12, 2008.

[59] M. A. Walker. Limited attention and discourse structure. *Computational Linguistics*, 22 (2):255–264, 1996.

[60] Lu Wang and Claire Cardie. Improving agreement and disagreement identification in online discussions with a socially-tuned sentiment lexicon. *ACL 2014*, page 97, 2014.

[61] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT-EMNLP)*, Vancouver, B.C., Canada, October 2005.

[62] Jie Yin, Paul Thomas, Nalin Narang, and Cecile Paris. Unifying local and global agreement and disagreement classification in online debates. In *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis*, pages 61–69. Association for Computational Linguistics, 2012.