D5.2 – Specification of Conversation Analysis / Summarization Outputs

Document Number	D5.2
Document Title	Specification of Conversation Analysis / Summarization Outputs
Version	1.3
Status	Final
Workpackage	WP5
Deliverable Type	Report
Contractual Date of Delivery	31.10.2015
Actual Date of Delivery	30.10.2015
Responsible Unit	USFD
Keyword List	Speech and social media summary defintion, addressing Period 1 review comment(s), initial speech and social media summarization approaches, future directions for Period 3
Dissemination level	PU

Editor

Ahmet Aker (University of Sheffield, USFD)

Contributors

Fabio Celli	(University of Trento, UNITN)
Morena Danieli	(University of Trento, UNITN)
Evgeny Stepanov	(University of Trento, UNITN)
Carmelo Ferrante	(University of Trento, UNITN)
Benoit Favre	(Aix Marseille University, AMU)
Balamurali A R	(Aix Marseille University, AMU)
Jeremy Trione	(Aix Marseille University, AMU)
Ahmet Aker	(University of Sheffield, USFD)
Rob Gaizauskas	(University of Sheffield, USFD)
Emma Barker	(University of Sheffield, USFD)
Monica Lestari	Paramita (University of Sheffield, USFD)
Marc Poch	(Websays)

SENSEI Coordinator

Prof. Giuseppe Riccardi

Department of Information Engineering and Computer Science

University of Trento, Italy

giuseppe.riccardi@unitn.it

Document change record

Version	Date	Status	Author (Unit)	Description
0.1	2015-07-31	Draft	Rob Gaizauskas (USFD)	Initial Outline
0.2	2015-08-04	Draft	Rob Gaizauskas (USFD)	Updates to Outline
0.3	2015-08-11	Draft	Ahmet Aker (USFD)	Writing up linking, clustering and summarization sections (Sections 3.1 - 3.5)
0.4	2015-08-11	Draft	Ahmet Aker (USFD)	Writing up Executive Summary
0.4	2015-08-11	Draft	Ahmet Aker (USFD), Fabio Celli (UNITN)	Writing summary section for the social media components (Section 3)
0.5	2015-08-20	Draft	Fabio Celli (UNITN)	Template-based Summaries (Section 3.6)
0.6	2015-08-20	Draft	Balamurali A R (AMU)	Cluster labeling (Section 3.4)
0.7	2015-08-27	Draft	Ahmet Aker (USFD)	Social media use case output refinements (Section 4.2)
0.8	2015-08-28	Draft	Ahmet Aker (USFD)	Introduction (Section 1)
0.9	2015-09-01	Draft	Benoit Favre (AMU) Ahmet Aker (USFD)	Editing Section 3
0.10	2015-09-01	Draft	Evgeny Stepanov (UNITN)	Editing Section 3
0.11	2015-09-01	Draft	Ahmet Aker (USFD)	Editing Section 4
0.12	2015-09-10	Draft	Benoit Favre (AMU)	Editing Section 3
0.12	2015-09-10	Draft	Ahmet Aker (USFD)	Editing Section 1
0.13	2015-09-14	Draft	Fabio Celli (UNITN)	Editing Section 3
0.14	2015-09-25	Draft	Benoit Favre (AMU)	Editing Section 3
0.15	2015-09-28	Draft	Ahmet Aker, Emma Barker, Rob Gaizauskas (USFD)	Editing Section 2.2
0.16	2015-09-28	Draft	Ahmet Aker (USFD)	Cross referencing social media definition

0.17	2015-09-28	Draft	Morena Danieli (UNITN)	Added speech summary definition
0.17	2015-10-01	Draft	Ahmet Aker (USFD)	Updating several sections
0.17	2015-10-02	Draft	Evgeny Stepanov (UNITN)	Updating results for speech use case
0.18	2015-10-02	Draft	Fabio Celli (UNITN)	Updated mood detection
0.19	2015-10-02	Draft	Rob Gaizauskas, Emma Barker, Monica Lestari Paramita, Ahmet Aker (USFD)	Added description of gold standard (Section 4.1) and Appendix
0.20	2015-10-08	Draft	Elisa Chiarani (UNITN)	Quality check completed. Some changes requested
0.21	2015-10-09	Draft	March Poch (Websays)	Review and requested changes
1.0	2015-10-14	Draft	Ahmet Aker (USFD)	Addressed review and quality check requests
1.1	2015-10-18	Draft	Rob Gaizauskas (USFD)	Proof reading; revisions to 4.5
1.2	2015-10-18	Draft	Ahmet Aker (USFD)	Adding further edits due to quality checks
1.3	2015-10-26	Final Version	Giuseppe Riccardi (UNITN)	Final checks

Excecutive Summary

In this report we do three things:

1. We specify what a good summary should contain for both the speech and social media use cases (this addresses reviewer recommendation 3 from the Period 1 review).
2. We describe the design and (intrinsic) evaluation of initial software conversation analysis and summarisation components developed to generate summaries for both speech and social media use cases.
3. We describe proposals for future refinements and extensions to our work on conversation analytics and summarisation, to be carried out in Period 3.

Contents

List of Acronyms and Abbreviations

Acronym	Meaning
Acronym	Meaning
THM	Town Hall Model
MCL	Markov Clustering
LDA	Latent Dirichlet Allocation
ACOF	Agent Conversation Observation Form
DECODA	Call center human-human spoken conversation corpus
LUNA	Spoken Language UNderstanding in MultilinguAl Communication Systems
QA	Quality Assurance
CRF	Conditional Random Fields
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
MMR	Maximal Marginal Relevance
DISCO	DIStributionally similar words using CO-occurrences
SCTM	Specific Correspondence Topic Model

1 Introduction

In this report we describe our follow up to period one activities in the SENSEI analytics and summarization work package (WP5). This includes definitions of what SENSEI's speech and social media summaries should contain, design and intrinsic evaluation of initial analysis and summarisation components and proposals for extensions and refinements to these components to be carried out in the project's final Period.

The report is structured as follows. First, in Section 2 we define summaries for both SENSEI speech and social media use cases. Sections 3 and 4 describe the initial conversational analysis and summarization modules for the speech and social media use cases, respectively. These sections include both a description of the algorithms used and, wherever possible, report results of initial intrinsic evaluation, as well as efforts undertaken to create resources for evaluation. Section 5 describes proposed future work to extend or refine the SENSEI analytics and summarization components in the project's final period. We conclude the report in Section 6

1.1 Follow-up to Period 1 Activities

For the speech use case, we have been working on two complementary abstractive approaches to synopsis generation. The first one (AMU) uses human expertize to create summary templates and fill them from entities detected in the conversations. The second one (UNITN) relieves humans from the effort of writing templates by generating them automatically from examples of synopses. The first approach is developped on the Decoda corpus while the second approach is established on the Luna corpus.

For the social media use case we developed two different approaches to generate summaries: an extractive approach (USFD) and a template-based approach (UNITN). Both summarization systems aim to provide a summary of comments users provide for news articles. The extractive summarization system performs a series steps including grouping comments into topic-based clusters, linking the comment clusters to segments within the news article to which they relate, ranking the clusters and finally extracting representative sentences from the clusters to include in the final summary. The outputs of the extractive summarization system were used in the extrinsic evaluation reported in D1.3. The template-based approach relies on topic extraction, mood prediction, agreement detection and metadata from the original article. Outputs of these sub-componenets are used to create an abstractive summary according to the Town Hall Meeting summary type reported in D1.2.

1.2 Follow-up to Recommendations from the First Review

In Section 2 we define summaries for both SENSEI speech and social media use cases. The definitions aim to adress the following reviewer recommendation:

recommendation n.3: The Consortium partners should define what constitutes a good summary within each task and formalize this definition for implementation in order to make it operational.

The summary definitions have been derived through experimental and user evaluations.

2 SENSEI's summary definition

In Section 2 we define summaries for both the SENSEI speech and social media use cases. This definition has been the basis for developing SENSEI's initial summarization components and also will inform refinements of the techniques developed so far.

2.1 Speech use case

For defining what is a speech summary, we may start from a general, and obvious, definition of summary such as the one proposed by Karen Spärck Jones in the late Nineties, i.e. "a reductive transformation of source text [...] through content reduction by selection and generalization of **what is important in the source**". If we apply this definition to the summarization of spoken conversations, the emphasised phrase within it refers to the complex issue of having criteria for deciding what is relevant and worth reporting when summarizing a dialogue between two (or more) speakers.

In SENSEI we flipped the issue, and we assumed that what is important in the source cannot be identified without taking into account the application task of the summary. For the speech use cases a summary is a reductive transformation of the source (spoken data) whose content may be represented in one out of a set of stereotypical reductive transformations.

In accordance with the SENSEI Reviewers' recommendation 3, we here provide a summary type definition, based on the speech use cases and user requirements. This definition provides a more detailed specification of the form a SENSEI speech summary should take.

In D1.1 we described four use cases where the SENSEI technology was applied with the aim of improving the daily job of potential users such as the Quality Assurance supervisors of the call centers. The use cases were designed on the basis of user requirements collected by interviews and focus groups with the users, from which we identified three stereotypical reductive transformations that are applicable in relevant contact center tasks. The different stereotypical types of summaries are outlined in Figure 1 below.

Conversation Oriented Summary: short synopsis of the content of the conversation.
Agent Conversation Observation Form (ACOF, henceforth): automatically filled questionnaires that aim to report relevant features of the behavior of call center agents, like their conversational competence, compliance to conversation guidelines, time call management, and empathy.

User-specified "ad hoc" reports: the opportunity of navigating the transformed conversations starting from queries specified by the users, like "see all the conversations where agents showed to have problems in using appropriated language"

Figure 1: Types of speech summaries.

During the second half of Period 1 of the project we focused on two speech use cases, i.e. ACOF and synopses generation, as reported in D1.2. The selection was done by balancing user needs with technical feasibility within the project. The key details of the selected use cases are presented in Figure 2.

Speech use case based on ACOFs
Users: Members of the Quality Assurance departments of call centre companies.
The User Goal: To be helped in assessing some behavioural skills and performance of the call center agents during inbound conversations with the customers.
The SENSEI Functionality: The system generates an ACOF, i.e. it assigns a value from a predefined set, including the values Pass, Fail,and Not Applicable, to the different questions of the ACOF. The ACOF includes questions that summarize the behavioural attitude of the agent during the call. The user may navigate the conversations processed by the system by asking questions like Find all the conversations where the agent failed in solving the customers issue, and get a report including the selected conversations and the associated ACOFs.
Speech use case based on synopses
Users: Members of the Quality Assurance department of call center companies.
The User Goal: To be helped in understanding the content of the inbound calls.
The SENSEI Functionality: The system generates a synopsis of the call focused on the call content more than on the agent's communication skills.
A **synopsis** is a summary that includes the following pieces of information:

- "why the customer called the call center?"
  - the main issues addressed in the call
- "was the agent able to solve the customer's problem and to deliver the required information?"
  - if and how the agent was able to satisfy the user requests
- "what issues did possibly emerge from the call?"
  - the issues where the agent did not solve the customer's problem either because the agent was not able to address it or the problem itself was impossible to be addressed during the call or in a single call.

Figure 2: Speech use cases from D1.2

2.1.1 Speech summary type definition

We now provide a more detailed specification of the types of summary based on the speech use cases outlined above.

Key content in ACOFs and synopses

- **Viewpoint in ACOFs:** the different items of the agent observation form express a view on different aspects of agent behaviour.
  - **A single agent behavior may be understood on the basis of multiple utterances of that agent throughout the call:** more than one utterance may need to be base of the pass/fail judge for a single item.

- **A single agent utterance may constitute the basis for judging more than one ACOF question.**

- **Viewpoint in synopses:** while the synopses are conversation oriented summaries (ideally, 7% in length with respects to the original call) and are focused on the (objective) content of the call, yet the issues reported, discussed, and solved in the conversations are usually expressions of the possibly different viewpoints of the agent and of the customer. The call center conversations often show an internal structure where the main issue (i.e. the problem that urged to be solved from the customer's viewpoint) originates sub-issues that may either contribute or prevent the solution of the customer's problem.

Given the considerations above, we propose that the summary of the call in form of ACOF expresses the viewpoint of the QA supervisor, and it ideally should report:

1. A Pass, Fail or Not-Applicable judge for each of the questions of the ACOF.
2. For Pass and Fail judges, the evidences on which the system based the classification, in form or transcription of the relevant conversation turns (or segment of turns).

A conversation oriented summary in form of synopsis expresses the viewpoints of customer and agent, and it should ideally report:

1. **The main issues of the conversation:** in call center conversations the main issues are the problems why the customer called; their identification constitutes the basis for classifying the call into several different classes of motivations for calling, like reporting hardware misbehaviour, asking information about train timetable, and so on. The main issue of a call should be prioritized for inclusion in the synopsis of the same call.
2. **The sub-issues in the conversation:** when in the conversation any sub-issue occurs, it may be there both because it is introduced by the customer or by the agents.
3. **The resolution of the call:** i.e. if the customer's problem was solved in that call (first-call resolution) or not.

2.2 Social media use case

In accordance with the reviewers' recommendation 3 we here provide a summary type definition based on the Social Media Use Case 1 (reported in D1.2). This summary type definition provides a more detailed specification of the form a SENSEI Social Media summary should take than the description given in D1.2. This summary type definition has inspired the initial social media components described in Section 4.

2.2.1 Background

In D1.2 we described various use cases for reader comment technology. Each use case specified a particular SENSEI functionality, which provides a particular view of the comment data to address a stated user goal. Use Case 1, which describes the scenario of a user interacting with a system to obtain an overview style summary of the discussion in the comments, was selected as the initial focus for summarization technology development. Key details of Use Case 1 are presented in Figure 3.

Users: Members of the public (news and comment reader, comment provider); News-media professional (reporter, sub-editor, comment editor).
The User Goal: To gain an understanding of the key content of a set of reader comments for a particular news article.
The SENSEI Functionality: The system generates a **Town Hall Meeting (THM) summary** - a short, mainly text-based summary, in the style of a town hall meeting news report ...
A **THM summary** will include the following information: (as if a reporter had asked the kind of questions he would use to cover a town hall meeting)

- "what were people talking about?"
  - the main issues addressed in the comments
  - the different views or perspectives on issues in the debate
- "how did people feel about the issues/what issues did people feel strongly about?"
  - what feelings were expressed on the issues? were there any intense feelings shown and on what issues?
- "what issues did people agree/disagree about?"
  - the issues where there was consensus / divided opinion
- "how many/what proportion of commenters were talking about the issues? shared such views? shared such feeling?'

This use case also describes how the summary will function as an index to the underlying comment: We provide links to the comments which contribute to the different parts of the summary to allow the user to **"drill down" to see the source comment(s)** in the context of the original thread/discussion. E.g., for each of the main topics listed, there will be a link to comments associated with that topic.

Figure 3: Use case 1.

Note there are a few minor differences in Use Case 1, as reported above, compared to Use Case 1, as described in D1.2:

1. We refer now to "issues" where we formerly referred to "topics and issues".
2. We have left out details referring to a summary of the article in the SENSEI output. (In this second period of the project it was decided that the priority for summarization should be to focus on the task of how to summarize comments associated with an article, but not the comments and the article. The challenge of summarizing the comments is both novel and difficult and therefore it made sense to concentrate resources on this problem, as opposed to diverting resource to the task of producing a summary for the article, which is a conventional single document summarization task).

3. We have left out details relating to who was taking part in the conversation.

2.2.2 Summary Type Definition

We now provide a more detailed specification of a reader comment summary based on Use Case 1 as outlined above. We first identify some key content or "ingredients" of reader comments. Then we define the type of content we require in a reader comment summary based on these ingredients. We also provide an example summary to illustrate how such content might appear in a reader comment summary.

We note that this is not intended as a formal content analysis of argument in comment; but it is sufficiently concrete to illustrate both the character of a reader comment summary and how such a summary relates to the source texts.

Key Content in Reader Comment

- **Viewpoint:** a comment may express a view on something (i.e. a comment often says something about something – an object or person, but more typically an event or a proposition).

  - **The same viewpoint may be expressed by multiple comments:** more than one comment may say the same thing about something.
  - **A single comment may express multiple viewpoints:** different views on things may be expressed within a comment.
  - **Viewpoints may stand in relation to each other:** i.e. one or more viewpoints may provide an alternative.

  Consider for example the following alternative viewpoints (in bold text) expressed in 3 comments posted in response to a proposal to change to a 3 weekly bin collection in Bury[1]:

  "I can't see how it won't attract rats and other vermin. I know some difficult decisions have to be made with cuts to funding, but this seems like a very poorly thought out idea".

  "It won't attract vermin if the rubbish is all in the bins. Is Bury going to provide larger bins for families or provide bins for kitchen and garden waste to cut down the amount that goes to landfill? Many people won't fill the bins in 3 weeks - even when there was 5 of us here, we would have just about managed".

  "I think it is an excellent idea. We have fortnightly collection, and the bin is usually half full or less. 2 adults and 3 children".

- **Other comment may relate to a viewpoint in a variety of ways, e.g.:**

  support a viewpoint with evidence; elaborate upon it; deny it or agree with it; clarify it; provide background to it; question it; consider the consequences of it, etc.

[1] http://www.theguardian.com/uk-news/the-northerner/2014/jul/17/rubbish-bury-council-votes-to-collect-wheelie-bins-just-once-every-three-weeks

(By “other comment”, we mean both (1) other content within a comment expressing a viewpoint and (2) other comments in the comment stream.)

For example:

“London’s awesome in the sunshine (viewpoint). Pubs are open, plenty of beer gardens if you know where to look, people out after work having fun, and generally good vibes all round (evidence/grounds for the viewpoint)”.

Summary Specification So, given this, what should a summary contain? To specify what a summary should contain we employ the abstract notion of **issue**. Essentially, an issue is a proposition about which multiple commenters express a viewpoint. We amplify this definition below.

Key characteristics of issues, as we use the term, are that:

(a) An issue is something that one can believe or disbelieve/agree or disagree with. For example “Climate change is directly caused by human activity”, “The US Senate should vote in favour of the Iran nuclear deal”, “Wheelie bins attract vermin”. Issues do not have to be significant; i.e. they need not be topics attracting front-page attention in the national press. They can also be simple statements about everyday issues such as “bus travel is free for the over 60s”. “Hard floors are better than carpets”. However, issues do have to be propositional: things like “climate change” or “the Iran nuclear deal” are not, abstracted from any context, issues since they are simply referential terms and do not on their own comprise an assertion that can be agreed or disagreed with.

(b) In the context of a comment set, issues are things that multiple commenters discuss, perhaps assert, deny, clarify, adopt alternative viewpoints upon, expand upon, qualify, consider the consequences of, etc. I.e. issues arise out of discussion/interaction and may be viewed as an expression of the commonality that underlies natural groupings of comments. Put another way, issues are abstract propositions that characterize the shared “aboutness” of a set of viewpoints. By “natural groupings” we mean the groups that human readers of comments tend to place comments into when asked to group them according to whether they are addressing the same thing.

(c) While issues are fundamentally propositional they may sometimes be expressed telegraphically by nouns or short phrases (e.g. “immigration”, “climate change”). However, for such abbreviated expressions to genuinely signify an issue it must be possible recover the underlying proposition from the context. For example, from previous comments it might be clear that in the comment “Climate change is a fabrication of the anti-business lobby” the expression “climate change” refers to the proposition “Climate change is caused by human activity” and that the comment expresses a viewpoint on that issue. An issue may also be thought of as something that can be expressed by a “whether or not” phrase or a yes-no question, to indicate opposing views in the comments e.g.: “whether or not climate change is directly caused by human activity”, “is climate change directly caused by human behavior?”.

(d) Note that not all comments relating to an issue need mention the issue explicitly. For example, in the context of the issue “less frequent bin collection will lead to an increase in vermin” a commenter might assert that “many people use compost bins without attracting vermin”. Clearly this comment addresses the issue yet it does not explicitly mention the issue.

(e) Sub-issues may emerge within the discussion around an issue, e.g. when evidence proposed as support for a viewpoint itself becomes the focus of discussion, attracting multiple comments perhaps contradicting or elaborating it. I.e. there is a recursive nature to issues and it can be rather arbitrary whether we deem something a sub-issue or not.

(f) Note also that threads and issues are not the same thing: comments addressing any one issue may occur in multiple threads and any one thread may contain comments relating to multiple issues. A comment replying to another need not be on the same issue and new issues or sub-issues can emerge as the discussion evolves into new areas. Intuition needs to be used in determining what seems like a new, distinct issue in the overall discussion.

Given this notion of issue, we can now offer our specification of what a content-oriented summary of reader comments should contain. **Ideally a summary should:**

1. **Identify main issues in the comments.** Main issues are those receiving relatively the most sizeable numbers of comment. They should be prioritized for inclusion in a space-limited summary.
2. **Characterise opinion on the main issues.** To characterise opinion on an issue typically involves:
   - identifying alternative viewpoints
   - indicating the strength of support for an issue or different viewpoints (aggregation)
   - indicating consensus or agreement among the comment
   - indicating disagreement among the comment
   - indicating qualitatively how opinion was distributed (e.g. using phrases like “Many said this; others said that”, “some said”, “most said”)
   - indicating evidence or grounds for a viewpoint
   - indicating whether the discussion was particularly emotional/heated and if so over what.

Figure 4 shows a summary containing these content elements. A number of “issues” are indicated – in red font. Characterisation of opinion on an issue is indicated in blue font.

For example consider Issue 1: “The majority of the comments discussed the climate of Britain and what constitutes a heatwave”. The text characterising opinion on this issue describes two alternative viewpoints:

“Some comments expressed disbelief at temperatures in the twenties meaning a heatwave for Britain and referred to temperatures elsewhere in the world and others reminded them that a heatwave is relative and that Britain tends to have a mild, temperate climate.”

This pattern is recursive: for more complex issues addressed by a substantial number of comments, a summary will ideally will include characterisation of opinion on the **different viewpoints** given.

For example, consider the text characterising opinion on issue 2, “how to stay cool in the heat”:

“The discussion ... mainly focused on air conditioning. Some felt air conditioning was very useful. One person said that they thought the UK would get hotter, and air con made it much easier to sleep. Some said that Air con units are unnecessary in the UK and too expensive to run. Several commenters said fans would suffice. A few comments detailed other ways of tolerating the heat without relying on air conditioning”.

- The text states that the majority of comment addressed the sub-issue signaled by the phrase air conditioning.
- Alternative viewpoints (on air-conditioning) are summarized:

  “Several commenters said fans would suffice”. “A few comments detailed other ways of tolerating the heat without relying on air conditioning”.
- The summary also characterizes opinion on this issue signaled by the phrase “air conditioning”:

  “Some felt air conditioning was very useful. One person said that they thought the UK would get hotter, and air con made it much easier to sleep. Some said that Air con units are unnecessary in the UK and too expensive to run.”

Issue 1: The majority of the comments discussed the climate of Britain and what constitutes a heatwave or **Issue 2:** how to stay cool in the heat. **Characterisation of Opinion – Issue 1 :** Some comments expressed disbelief at temperatures in the twenties meaning a heatwave for Britain and referred to temperatures elsewhere in the world, while others reminded them that a heatwave is relative and that Britain tends to have a mild, temperate climate. **Characterisation of Opinion – Issue 2:** The discussion of **Issue 2:** how to stay cool, mainly focused on air conditioning. Some felt air conditioning was very useful. One person said that they thought the UK would get hotter, and air con made it much easier to sleep. Some said that Air con units are unnecessary in the UK and too expensive to run. Several commenters said fans would suffice. A few comments detailed other ways of tolerating the heat without relying on air conditioning.
Issue 3: Some comments expressed negative opinions about hot temperatures. **Charcterisation of Opinion – Issue 3:**A small group of comments discussed how the heat brings about the nuisance of midges and how to deal with them.
Issue 4: Several comments discussed how newspapers report on weather in Britain, **Characterisation of Opinion – Issue 4:**with commenters in agreement that headlines were predictable and many feeling that such articles were not necessary.
Issue 5: Several comments discussed the weather and living in London in the summer, **Characterisation of Opinion – Issue 5:**with opinion fairly divided on whether or not it was a positive experience.

Characterisation of Opinion – Issues 1-5: Much of the discussion was not serious with many humorous comments and jokes.

Figure 4: Example summary containing content elements.

3 Initial Conversation Analysis/Summarization Outputs for Speech

In this section we report two complementary abstractive approaches to synopsis generation developed for the speech use case.

3.1 Hand-written template-based abstractive summarization of call-centre conversations

This section describes the abstractive summarization approach developed for the Decoda corpus. The approach is an extension of template-based summarization which consists in generating a template from a cluster of conversations, and then filling that template with named entities and other information gathered from the conversation to be summarized.

3.1.1 Algorithm

Call-centre conversation synopses are short summaries of the events taking place during a conversation between a caller (or user) and one or more agents. Such a synopsis should contain a description of the user need or problem, and how the agent solves that problem. It might describe as well the attitude of the caller and the agent. Table 1 shows a few examples of synopses from the Decoda corpus. In addition, Figure 5 contains an example of a transcript from a conversation. These examples clearly show that abstractive approaches are required for call-centre conversation summarization.

Agent: ⟨name⟩ hello
Caller: yes hello
Agent: hello madam
Caller: are buses uh 172 and 186 running?
Agent: unfortunately on the 172 and uh 186, we got the information this morning, there's a notice from the B depot in Vitry so it was known uh yesterday evening and this morning
Caller: uh yes
Agent: so the buses are very disrupted uh this morning huh some uh, uh have gone out and others not, so there are very major disruptions on these two bus lines huh
Caller: whew that's really irritating because what will people who are working do
Agent: unfortunately yeah, it's annoying huh I understand that uh actually
Caller: further there was a notice that was uh
Agent: frankly not uh
Caller which in fact creates in the private... who risk their post, if they're not going to work because those gentlemen have decided to strike
Agent: it's me I somewhat agree with you
Caller: someone from the RATP who agrees with me...

Figure 5: Extract from Decoda conversation (20091112-RATP-SCD-0042) translated from French.

Table 1: Example of synopses from the Decoda corpus including annotator ids (translated from French).

Conversation	Ann.	Synopsis
20091112-RATP-SCD-0042	1	Are buses 172 and 186 running? No, disrupt because of Vitry depot strike, complaint and compassion
20091112-RATP-SCD-0042	2	Query of information on the status of buses 172 and 186. Major disruption on these lines due to a strike. Complaint from the caller.
20091112-RATP-SCD-0604	1	Suburban itinerary, tryed by the RATP website but not convinced of the proposed route. Summary of the route by the caller.
20091112-RATP-SCD-0604	2	Itinerary request to go from Chilly-Mazarin station from Fontenay-sous-bois station. Take the RER A towards Saint-Germain en Laye until Gare de Lyon, then metro 14 towards Olympiade, get off at Bibliothque, then take RER C to Chilly Mazarin. Communication of the trip duration and frequency of trains.
20091112-RATP-SCD-0604	3	A woman is phoning cause she wants to ask for an itinerary for another person. It is a bit confusing cause there a many changes but in the end it seems to be clear.
20091112-RATP-SCD-0285	1	Request for itinerary suburbs to Paris to understand the fare given by the caller employee.
20091112-RATP-SCD-0285	2	Request for information about the zones to take for a Navigo card for one person living in Chailly-en-Brie to travel in Paris. Zones 1 to 6.
20091112-RATP-SCD-0285	3	An employer is phoning the customer service cause he is not very sure about the ticket he has to pay for his employee. His employee is asking him for a sum which doesn't correspond to the fares and so he has the feeling that he is being ripped off.

The Decoda corpus consists of recordings at the RATP call-centre from a two-day period during strikes in 2009. Topics covered by the conversations include traffic information (status of the lines under strike), itinerary search, schedule requests, lost-and-found, fares and monthly passes, etc. Given a conversation falling in one of these topics, it is often straightforward to extract relevant slots and manually devise a template that would fit human-written synopses. This is the heart of the approach proposed for abstractive synopsis generation.

The approach consists of 3 steps:

1. Template matching: find which template or set of templates to use for a given conversation
2. Named entity detection: find evidence from the conversation for filling the slots in the templates
3. Slot filling: chose from evidence which pieces of information best fit each slot

The first step (template matching) consists in finding the best template for an input conversation. It can be reasonably assumed that one template has been created for each of the topics, so that the problem reduces to topic classification, a well-studied problem for call-centre conversation transcripts. Even if that assumption was not realistic, one could use the following baseline for matching a conversation with a template: from all training data conversations, match them manually with one of the templates, then at test time use the template associated with the closest training conversation from the test conversation in term of a conversation similarity metric (such as bag-of-word cosine similarity). Topic classification performance has been shown to reach 86.5% on the Decoda corpus by [1], using factor analysis and other methods leveraged from the speaker identification community. The Decoda topics are listed in Table 2.

Table 2: List of main conversation topics annotated in Decoda.

Freq	Code	Description
342	ETFC	Traffic
263	ITNR	Itinerary
242	OBJT	Lost&Found
173	NVGO	Navigo pass
70	HORR	Schedules
69	AAPL	Other call
69	PV	Tickets&Fines
57	NULL	None
51	VGC	Large groups
46	TARF	Fares
38	OFTP	Transportation offers
25	RETT	Late bus
21	CPAG	Agent behaviour
12	ACDT	Accident
12	JSTF	Justification
11	SVDO	Customer care
9	RGLM	Payment
5	DDOC	Query for documents
4	AESP	Passenger bahviour
3	ACCS	Accessibility
2	SRTP	Website / mobile

The second step (named entity detection) consists in finding the named entity or piece of information candidates to fill slots of the template. There exists a large body of work on named entity recognition, concept detection and slot filling in speech transcripts. Most approaches to named entity recognition rely on Conditional Random Fields trained to categorize each word in begin-inside-outside labels for the entity categories (persons, places, organizations...) reaching accuracies around 89% F-score on newswire [2]. Recent approaches based on deep neural networks, such as SENNA [3] yield similar performance. Named entities from the Decoda corpus are listed in Table 3. For specific pieces of information required for filling templates, it is not possible to rely on named entities. For instance, in the Lost&Found scenario, the type of object lost, and the fact that the lost object was found or not are not conveyed by any named entity type. Therefore, we extend slot candidates to any non-named entity annotations generated by the systems from WP3 and WP4 (semantic parsing, para-semantic analysis and discourse parsing). In the Lost&Found scenario, we would use the `Lost_Item` frame element (patient) of the `Losing` frame for the lost object. The fact that the object was found would be tracked throught the `Finding` frame.

The final step (slot filling) consists in filling the slots with appropriate entities and lay the template sentences to generate the final synopsis. The problem is that entities have been categorized with a broad type (such as location) whereas the template requires finer-grained types (such as departure-location, arrival-location). A baseline for matching entities with slots is to filter them by type and then

Table 3: Named entities annotated in the Decoda corpus

Frequency	Symbol	Description
12756	A	Address
4778	T	Mean of transportation
3700	HU	Person
2316	H	Time
1895	ORG	Organization
1697	PRD	Product
653	P	Price
563	TEL	Telephone
419	D	Date
151	CP	Postal code
121	L	Location
51	M	Monument
7	IMT	License plate

use them in order of appearance (assuming the templates have been crafted to roughly support that natural order). Then, two approaches can be devised for improving over that baseline: One can project the finer labels to the conversations through hand-written synopsis alignment with templates and train a named entity recognizer which supports fine-grained labels, or another option is to devise rules based on conversation content around each entity for selecting its subtype.

In addition to those steps, a set of templates have to be created for each topic. During this period of the project, we focused on manual template writing. The process for creating templates consists in gathering reference synopses for a single topic, analysing and extracting recurrent core semantic frames in these synopses, and writing generic template sentences with slots in place of relevant entities. Some parts of the templates are optional in order to account for variability in conversations. Table 4 gives examples of templates for a subset of the topics. The templates have been defined as regular expressions with grouping, quantifiers (question mark for zero or once, star for Kleene closure), alternatives (pipe is the alternative operator) and slot variables (upper caps preceded by dollar). Each template is split in sentences which can be generated independently, after all their slots are fulfilled. For a given conversation, the system can draw from multiple templates according to the detected topics. In particular, the "Other" template is a catch all for sentences which might be used in various topics. Slot variables are instanciated from evidence garnered from the conversation, such as named entities, frames, etc. Slot names are chosen globally so that slots from different templates can benefit from cross-training. The order of the sentences is computed from the order of the slots detected in the conversation.

We have devised two strategies for filling the slots: define rules for picking the evidence from the conversations, and annotated training conversations with slots and train a classifier to predict them.

The first strategy consists in associating each slot with a set of rules based on conversation annotations for finding the slot value. For instance, a $FROM slot can be filled with a location named entity governed by "from" in the syntactic dependency tree, or which is the `Source` argument of a `Travel` frame. When multiple word strings match the set of rules, the first one occuring in the conversation is taken. When the same slot can be filled multiple times in the template, such as when using a Kleene closure, they

Table 4: Templates for select Decoda topics (translated from French).

Code	Topic	Template
HORR	Schedule	Query for schedules (using $TRANSPORT)? from $FROM to $TO.
ITNR	Itinerary	Query for itinerary (using $TRANSPORT)? from $FROM to $TO (without using $NOT_TRANSPORT)?. (Take the $LINE towards $TOWARDS from $START_STOP to $END_STOP.)*. Query for location $LOCATION.
NVGO	Navigo pass	Query for (justification|refund|fares|receipt) for $CARD_TYPE. Customer has to go to offices at $ADDRESS.
OBJT	Lost&found	$ITEM lost in $TRANSPORT (at $LOCATION)? (around $TIME)?. (Found, to be retrieved from $RETRIEVE_LOCATION |Not found).
TARF	Fares	Query for fares from $FROM to $TO. The fare is $COST.
ETFC	Traffic	Query for state of line $TRANSPORT. (Frequency is $FREQUENCY |Not running because of $ISSUE |Cannot get information because of $ISSUE)
-	Other	Call corresponding service at $PHONE_NUMBER. Send a mail query to $ADDRESS.

are matched in order.

The second strategy assumes that conversations are annotated with slot segments in order to be able to train a sequence classifier such as CRFs. Such annotation is expensive on the scale of a call-centre corpus such as Decoda. Therefore, we devised an approach for lightly supervised learning of the classifier. The idea is that it is relatively straightforward to annotate reference synopses with slot variables manually, and then we can map the synopsis slot values to the original conversation using mapping functions based on automatic annotations of both the conversations and the synopses. A benefit of this approach is that it also yields a reference for slot values which can be used to evaluation the qualtity of the slot filling component, and it also helps assessing how much of the reference synopses are covered by the templates.

The synopsis-conversation matching algorithm works as follows:

- function match_synopsis() // for a synopsis hand-annotated with slots:
  - For each slot value:
    - For each sentence of the corresponding conversation:
      - Align the slot with the conversation using Lenvenstien alignment with a custom cost function
- function substitution_cost() // substitution cost in Levenstein alignment
  - return 0 for an approximate match of the words
  - return 0.1 if one of the words is a stopword
  - return ∞ for other content words
- function inserstion_cost() // insert a synopsis word

  - return 1 if the word is a stopword
  - return ∞ for other content words
- function deletion_cost() // insert a conversation word
  - return 0.1 if the word is a stopword
  - return 1 for other content words

For approximate word match, we normalize accents of the words and lowercase the characters, and allow words to be at a character-level Levenshtein distance of 1 when they are sufficiently long. This alignment allows to match 85.57% of the synopses slots back to the conversations. The remaining mismatches are mainly due to errors in the annotation, or general concepts that cannot be matched to words. We leave this for future work.

We performed a preliminary experiment in order to train a CRF to recognize those slots using the following features: word (up to 3-gram), part-of-speech (up to 3-gram), lemma, named entity label, governor word, governor part-of-speech, dependency label. Slot labels use a begin-inside-outside coding. The system is trained with CRF++ and performs at a 16.58% F-score. Careful analysis of the results show that the system has low recall on highly lexicalized slots for which it does not have much training data. On those slots for which it can generalize ($ITEM, $TO and $TRANSPORT), the the F-score is higher, around 50%. Clearly, slot filling for synopses is not a task that can be sloved with local features, and the classifier needs to be able to handle conversation-level features.

Therefore, we performed another experiment in which we try to classifiy each noun phrase of the conversations according to its slot label or no slot label (16 slot labels + 1 no slot class). We train the icsiboost classifier to predict those labels according to the following features extracted on the phrase: head word, part-of-speach, lemma, and named entity type, governor word, part-of-speech and dependency label, phrase word and part-of-speech bag, occurrences of named entity type and head lemma since start of conversation, conversation topic, phrase length, relative position of phrase in conversation, speaker role (caller / agent). The classifier is trained with 100 rounds of boosting on the CCCS training data and obtains an error rate of 13.06% on the 17 classes. Most relevant features are the topic, the occurrencies of the named entity type and the lemma, the named entity type of the head, the head lemma and the phrase bag of words.

In order to generate slot values, given a conversation and the associated template, we use the highest scoring phrase for that slot in the conversation, if its posterior probability is higher than 0.1.

3.1.2 Evaluation

We performed evaluation on a subset of templates on the CCCS test set[2]. using the ROUGE-2 evaluation metric. Extrinsic evaluation will be performed with call-centre supervised as part of WP1 during Period 3.

First, we setup two baselines and one topline. The first baseline consists in replacing the slot values with a bogus token which will not be matched by Rouge during evaluation. This baseline defines the

[2] See the description of the CCCS shared task organized at Multiling'15 in D7.4.

worst score one can every get from using the templates. The second baseline is based on the assumption that named entities play an important role in synopses: it consists in concatenating conversation named entities until the length constraint, without repetition. This baseline achieves a very bad readability, as expected. The topline consists in replacing the slot values with those manually annotated in the reference synopses. For reference, we also give extractive baseline scores on the same dataset. Results are summarized in Table 5. The rule-based and icsiboost systems are defined and trained according to the approaches defined in the previous section.

Table 5: Rouge-2 results of the Decoda synopsis generation systems on a subset of the CCCS test set.

Type	**System**	**Rouge-2**
Extractive baseline	Longest turn	0.06419
Extractive baseline	Longest turn @ 25%	0.07816
Extractive baseline	MMR	0.07007
Abstractive baseline	Templates with bogus slots	0.02228
Abstractive baseline	Named entity concatenation	0.09337
Abstractive system	Templates with rule-based slots	0.10084
Abstractive system	Templates with icsiboost slots	0.10443
Abstractive topline	Templates with reference slots	0.18067

Results show that the abstractive methods yield better results than extractive ones and the baselines, even though Rouge is not a metric particularly favorable to abstractive approaches (since it cannot handle synonyms). The only baseline to achieve good Rouge score is the concatenation of named entities, at the price of poor readability which makes it hard to make any sense from them. The slot prediction system is very promising as it works better than the system with simple hand-crafted rules. Nevertheless, achieving the topline is still far ahead and will require more effort in extracting better features (using more annotations from other WPs) and mapping slot values from annotated synopses to conversations.

Here are a few examples of synopses for which the template filling system worked well:

- *Le client a perdu son sac dans le RER entre Cergy et Poissy.* (The client lost is bag in the RER between Cergy and Poissy)
- *L'appelant voudrait se rendre de trois rue d' Alger Massy. Le conseiller lui dit de prendre le RER B jusqu' Massy.* (The caller would like to go to 3 rue d'Alger in Massy. The agent tells him to take the RER B to Massy.)
- *Demande d'horaire pour se rendre Croix de Berny. Prochain passage treize heure trente.* (Query for schedules to go to Croix de Berny).

This list gives examples where the system did not predict the correct slot values:

- *L'appelant voudrait se rendre de arrłt Fischer station Fischer. Le conseiller lui dit de prendre le ligne jusqu' station Fischer.* (The caller would like to go from the Fischer stop to Fischer station. The agent tells him to take the line to Fischer station — repeated named entity, missing line name)

- *L'appelant voudrait se rendre de gare de Drancy Drancy.* (The caller would like to go to Drancy station in Drancy — did not find an itinerary)
- *L'appelant voudrait se rendre de rue Taclet rue Taclet. Le conseiller lui dit de prendre le cent deux jusqu' rue Taclet.* (The caller would like to go to Taclet Street at Taclet Street. The agent tells him to take the 200 until Taclet Street. — repeated named entity)
- *L'appelant voudrait se rendre de Denfert-Rochereau trente six rue d'Assas. Le conseiller lui dit de prendre le ligne jusqu' Paris.* (The caller would like to go from Denfer-Rocherau to the 36 Assas street. The agent tells him to take the line to Paris. — correct query but wrong problem resolution)

An analysis of the incorrect synopses shows that the approach can sometimes mistake multiple occurrences of the same entity for different slots. This can be fixed by taking advantage of coreferences. Another problem is that of unfilled slots. This will take more work so that recall increases. In addition, the approach does not account for unexpected situations not handled by the templates. The particular problem is explored in the next section.

3.2 Template generation for abstractive speech summarization

3.2.1 Algorithm

The UniTN approach to abstractive summarization of call center conversations follows the approach of [4] and adapts the system to SENSEI data and languages. The system consists of several components, specifically:

- Template generation from manual summaries/synopses
- Topic segmentation for conversation transcripts
- Key entity/phrase extraction & template-slot population

The algorithm is supervised and training relies on linking the conversation utterances to the summary sentences. In LUNA Corpus such annotation is not available. Thus, several linking heuristics were employed:

1. Entire conversation is linked to the summary;
2. Top 4 utterances based on the Cosine Similarity;
3. Top 4 utterances based on the Cosine Similarity after the generalization of conversation and summary terms to WordNet synset IDs;
4. Top 4 utterances computed and compared as average vector of all words in a turn and synopsis using word2vec vectors;

Table 6: Abstractive summarization performances reported as ROUGE-2. The lower half of the table reports performances for utterance to synposis sentence linking heuristics.

Model		EN	IT
Extractive Summarization			
Baseline	Longest Turn	0.024	0.015
Baseline	Longest Turn from Beginning	0.025	0.027
MMR		0.030	0.020
Abstractive Summarization			
Template	*All Conversation*	0.021	0.018
Template	*String CosSim*	0.041	0.021
Template	*Synset ID CosSim*	0.042	0.025
Template	*Word2vec similarity*	0.056	0.029

In the template generation step word are replaced by Synset IDs from the WordNet. The resulting summaries are clustered and then they are clustered to form 'generalized' templates. The module depends on various Natural Language Processing tasks: Named Entity Recognition, Part-of-Speech tagging, Shallow Parsering (chunking), and Dependency parsing.

In the summary generation step, a conversation is segmented with respect to topics and key entities/phrases are extracted. The extracted entities are used to populate the generalized templates (i.e. synset IDs are replaced). As a final step, the filled templates are ranked using a language model.

3.2.2 Evaluation Results

Table 6 reports evaluation results using the ROUGE-2 evaluation metric, comparing the performances to extractive summarization baselines from the MultiLing 2015 Shared Task.

4 Initial Conversation Analysis/Summarization Outputs for Social Media

In this section we describe initial social media components developed to address the first use case scenario for social conversation analysis and summarization, namely the Town Hall Summary. Details of the Town Hall Summary are given in Section 2.2. Our initial components resulted at this stage in two different summaries: an extractive summary and a template-based summary.

Extractive summarization works as a three stage process. First it groups the poster comments into clusters. Each cluster is intended to contain comments talking about the same or similar topic or issue. Next, it divides the comment clusters into linked and unlinked classes. Linked comments/clusters are those referring to or discussing topics within the news article. Unlinked clusters discuss topics not entailed in the article. Finally, different summaries are produced based on the outputs of the first two stages. The three components that build extractive summaries are described below in Sections 4.2, 4.3 and 4.5. Note we also provide labels for the clusters created in the first step. The cluster labeling module is described in Section 4.4. Also note that the outputs of these different modules (including the cluster labels) have been used in the extrinsic evaluation carried out by USFD for the social media use case and reported in section 2.3.3 of D1.3; section 3.3 of the same deliverable reports the results of the evaluation. Section 3.3.1 of D6.2 presents the SENSEI user interface, which uses the outputs of the components detailed here, and explains the technical details behind the UI.

Template-based summarization uses data from three different modules (topic extraction, mood prediction and agreement/disagreement detection) and metadata from the article (extractive summary). The template has been designed following the Town Hall Meeting use case. The template-based summary includes: an introduction with the title and subtititle of the article, a list of the most frequent issues or topics, a development, with the emotions and consensus associated to topics, and a conclusion, with the most active contributor. The template is filled in 4 stages: 1) the system extracts the issues or topics from article and comments with LDA, 2) the system predicts agreement/disagreement and mood for each comment and 3) the system matches topics, agreement/disagreement and mood at comment level. Finally 4) the system computes the rank of most active contributors. The template-based summarization is described in Section 4.6.

4.1 Creating a Gold Standard for Intrinsic Evaluation

Before describing the software components for extractive summarisation that we have developed and evaluated, we first describe the resource we have created for intrinsic evaluation.

While there are several approaches one might adopt to evaluate clusters and summaries of reader comments, we believe the optimal approach is to have a human-authored gold standard. A gold standard resource is valuable for a number of reasons: i) it should provide example annotations – in this case summaries and clusters – that may give insights into system development. (Example summaries are especially valuable in such a novel domain where there are so few summaries available) ; ii) system outputs may be compared against the Gold Standard and scored either automatically, using a metric like ROUGE [5] or manually, in the case of summaries using a shared content identification approach like the Pyramid method [**?**]; iii) finally, Gold Standard data can provide development data to train or tune parameters in machine learning approaches.

To date, to the best of our knowledge, there is no collection of human-authored informative summaries of reader comment. Nor is there a collection of human gathered clusters of topically-related comments. This poses a real challenge for technology development in this area.

4.1.1 Aims

Our aim was to create a collection of human authored summaries, which conform to the summary definition given above in section 2.2. There were two reasons to do this: (1) to determine that humans could indeed author summaries, by no means a straightforward task given the complexity of the summary definition in relation to the source text (2) to create a resource to be used for intrinsic evaluation of clustering and summarisation technologies.

We now describe the method we developed for comment summary writing and the resulting corpus of summaries and related annotations.

4.1.2 Method

Anyone wishing to create such a collection of summaries for reader comment must address the substantial difficuties of comment volume and complexity. Comment streams typically contain at least 100 comments and often total in excess of 1000, and as we have seen above, similar views can be expressed in multiple comments, within a thread or in many threads while multiple different viewpoints may be expressed within a single comment. Moreover the dialogic nature of comment means ellipsis and anaphor are common; often a comment may comprise just a single token, (to express an emotion or opinion), yet we can also find examples of lengthy monologues on an issue or set of issues.

How might an annotator begin to make sense of such comment, to identify issues and characterise opinion?

To help people write good summaries which match our criteria in a consistent manner, we have developed a novel method in 3 stages: comment labeling, label grouping and summary generation. In a fourth stage, "back-linking", we ask people to link summary sentences to supporting groups (the information from back-linking is useful for evaluation purposes, but is not part of the summary writing process). We provide support for each stage in a specially designed interface for writing comment summaries (see section 4.1.3 below). Further details for the summary writing task and the stages of labeling, grouping, summary generation and "back-linking" follow.

Summary Writing–Introduction for Annotators The aim is to provide a general purpose, overview type summary of a set of comments. They should assume that a good summary is one that can be understood independently by someone who has no detailed knowledge of the content being summarised. We do not ask annotators to summarise the article. (Options for presenting the article context include: i.) we display the summary in the context of the full article, ii) add in an article summary automatically, a simple option being to take the first n lines of the article, so for example a summary may be preceded by the line "In response to an article about [first line of the article] the comments said...".)

We provide annotators with an example summary and article to indicate the type of summary we expect; but emphasis is placed on guidance for labeling, grouping, and summary generation.

Labeling A label should record the main "points, arguments or propositions" expressed in the comment. In many respects we can view a label as a summary of an individual comment.

Example:

We do not insist on the precise form a label should take – we encourage the use of a personal shorthand – but we advise on the following:

- To use common keywords and abbreviations, to associate similar content in different comments.

- To make a note of any grounds or evidence a comment gives in support of a proposition or argument.
- To make notes when a comment agrees or disagrees with something/someone.
- To make a note of jokes, strong feeling and other emotional content if its is a strong feature of the comment.
- To try and label all comments-especially very brief ones
- To make explicit embedded content, e.g. in ellipses, which may indicate support for particular arguments or propositions, example: a very brief comment, simply agreeing with something said before and a label which records this position. (thus labels can filter and reduce a comment but can also make content explicit).
- To be aware of patterns of interchange (e.g. the same person repeating a point to several others).

Labelling is a detailed process, which if done properly takes time, but there are two main benefits: First, a label records the essential content of the comment in a form which is easier to interpret and manipulate. Thus the annotator prepares the *data* for the processes of “grouping” and “summary generation”. Second, labelling helps a person to engage with the comments in a systematic fashion. By thinking carefully about what to select for a label, annotators should gain a good understanding of content and are more likely to remember details. (Useful e.g. for recalling examples.) In sum the *annotator* is better prepared for “grouping” and “summary generation”.

Grouping In stage 2 we ask annotators to sort and group together related labels. This builds on the process of using common expressions to associate labels and content (see stage 1). Grouping is a carried out in the context of writing a comment summary – the annotator’s goal in grouping is to make sense of the discussion and in preparation for writing. People should *not* be guided by ideas about what might be useful for system development or evaluation (although we may use annotations for such ends).

We restrict grouping to 2 levels: Group and Sub-Group. Labels may be copied and placed in multiple Groups/Sub-Groups, as required. We advise annotators to create a Group/Sub-Group Label to summarise the content in a Group/Sub-Group in terms of e.g. topics, arguments or propositions, humour, strong feeling and different viewpoints.

Label metadata (we gather labels and provide the following for convenience) includes the comment id (based on its position in the 100) and the comment provider id. An annotator may use this information to count the number of commenters contributing to different discussion points, to follow patterns of interchange between comments and to track approximately how issues develop over time. In Grouping, annotators work mainly with the labels, but they may revisit source comments, in the context of the comment stream, if they wish. They may also edit labels further, in stage 2, when necessary.

We do not set constraints on grouping in terms of time or number. But all labels should be placed in a Group (we include a ”miscellaneous” Group). When an annotator judges that he/she has sorted and characterised the data sufficiently for the purpose of writing a summary, they proceed to stage 3.

Summary Generation Providing a view of the results of Grouping and Labeling and the original comments, we ask annotators produce a summary in two different lengths (the summary should be based on their analysis in the preceding stages). Unconstrained summary: a summary, of a comfortable length, written with the knowledge that the final aim is to produce a summary of around 150-250 words. Constrained summary: a summary of restricted length, minimum 150 to maximum 250 words.

While group labels play a key role in formulating the summary, further analysis and content selection may take place as a person decides on the text to include. When formulating summary sentences an annotator may:

- Develop a group label further, e.g. adding further information based on labels in the group or producing a polished or complete sentence.
- Carry out further grouping or sub-grouping in their head – e.g., the most effective way to summarise content from two separate groups might be to create a new high-level statement based on the two groups.
- Exemplify, clarify or provide background to an expression, using details from labels or comments within a group.

We do not provide detailed instructions on how to order content, but the summary should be well formed and readable.

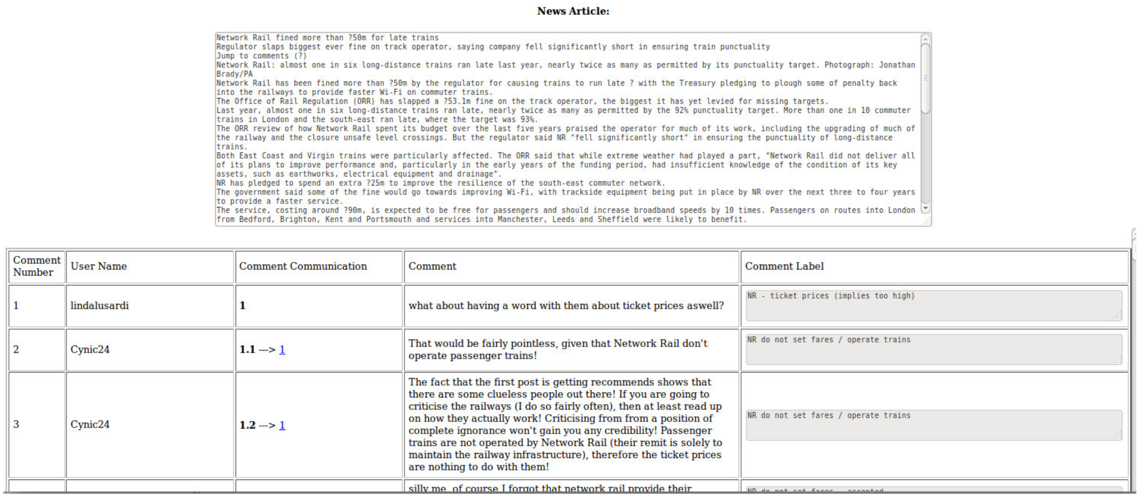
We encourgae annotators to provide rough estimates of the quantity of comments talking about the various topics and issues; and in terms of the extent of opinion and feeling shown toward different issues. Annotators may count labels in the groups that inform their summary, and then use an appropriate quantifier, (but we do not insist on an exhaustive or precise count of labels/comments in the summary). We recommend examples for quantifiers such as:

- Most of them comments
- Them majority of comments
- Opinion was divided
- At least
- In the minority
- Outliers
- Exceptions
- A few
- Several
- All

- Many
- A substantial number of
- Some, etc.

Stage 4: Back-Linking summary sentences to Groups We ask annotators to link each summary sentence (using the saved, constrained length summary) with the Group(s) of labels, and/or Subgroup(s), that informed or supported that sentence.

If a sentence was based on information in multiple Groups they should link to all relevant Groups. Linking to a Group implies that at least some of the labels in that Group have played a part supporting the sentence, (i.e. not all labels in a Group need to have played a role for their to be a link). Linking to Sub-groups: if the labels that support a summary sentence are localized to one or more Sub-groups, then only those specific Subgroups should be selected. Otherwise, the entire Group should be selected. Annotators should not select both a Group and Sub-groups within a group for linking to a sentence (selecting the Group implies that all Sub-groups have potentially played a part).



News Article:

```
Network Rail fined more than ?50m for late trains
Regulator slaps biggest ever fine on track operator, saying company fell significantly short in ensuring train punctuality
Jump to comments (?)
Network Rail: almost one in six long-distance trains ran late last year, nearly twice as many as permitted by its punctuality target. Photograph: Jonathan
Brady/PA
Network Rail has been fined more than ?50m by the regulator for causing trains to run late ? with the Treasury pledging to plough some of penalty back
into the railways to provide faster Wi-Fi on commuter trains.
The Office of Rail Regulation (ORR) has slapped a ?53.1m fine on the track operator, the biggest it has yet levied for missing targets.
Last year, almost one in six long-distance trains ran late, nearly twice as many as permitted by the 92% punctuality target. More than one in 10 commuter
trains in London and the south-east ran late, where the target was 93%.
The ORR review of how Network Rail spent its budget over the last five years praised the operator for much of its work, including the upgrading of much of
the railway and the closure unsafe level crossings. But the regulator said NR "fell significantly short" in ensuring the punctuality of long-distance
trains.
Both East Coast and Virgin trains were particularly affected. The ORR said that while extreme weather had played a part, "Network Rail did not deliver all
of its plans to improve performance and, particularly in the early years of the funding period, had insufficient knowledge of the condition of its key
assets, such as earthworks, electrical equipment and drainage".
NR has pledged to spend an extra ?25m to improve the resilience of the south-east commuter network.
The government said some of the fine would go towards improving Wi-Fi, with trackside equipment being put in place by NR over the next three to four years
to provide a faster service.
The service, costing around ?90m, is expected to be free for passengers and should increase broadband speeds by 10 times. Passengers on routes into London
from Bedford, Brighton, Kent and Portsmouth and services into Manchester, Leeds and Sheffield were likely to benefit.
```

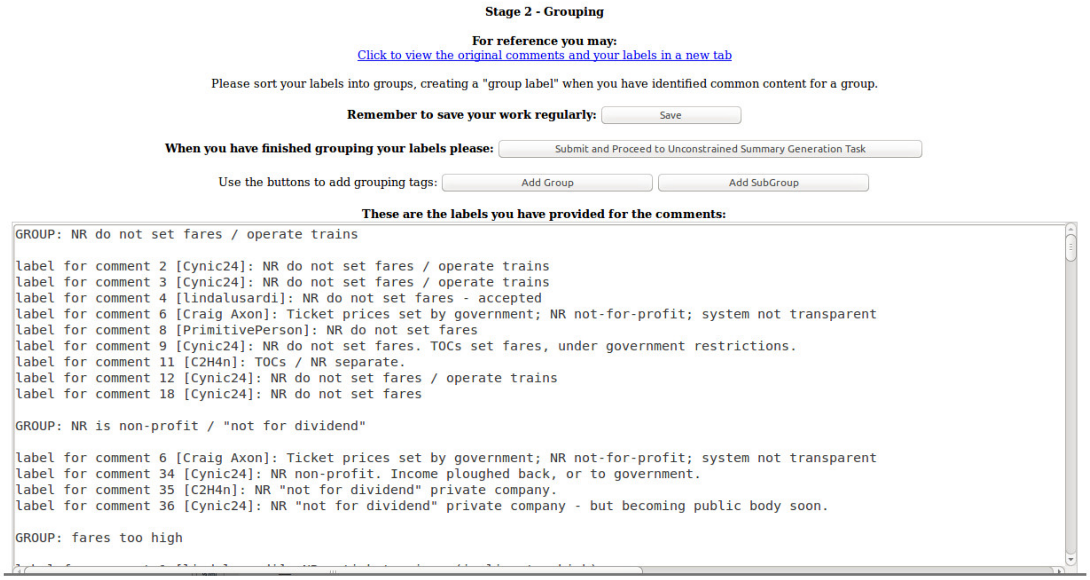
Comment Number	User Name	Comment Communication	Comment	Comment Label
1	lindalusardi	**1**	what about having a word with them about ticket prices aswell?	NR - ticket prices (implies too high)
2	Cynic24	**1.1** ---> 1	That would be fairly pointless, given that Network Rail don't operate passenger trains!	NR do not set fares / operate trains
3	Cynic24	**1.2** ---> 1	The fact that the first post is getting recommends shows that there are some clueless people out there! If you are going to criticise the railways (I do so fairly often), then at least read up on how they actually work! Criticising from from a position of complete ignorance won't gain you any credibility! Passenger trains are not operated by Network Rail (their remit is solely to maintain the railway infrastructure), therefore the ticket prices are nothing to do with them!	NR do not set fares / operate trains
			silly me, of course I forgot that network rail provide their	NR do not set fares - accepted

Figure 6: Window for Labelling Comments.

4.1.3 Tool support

To support annotators in the summary writing task we have developed an interface that i) allows an annotator to select an article and comment set for anno- tation, then ii) displays the article and the associated set of reader comments, preserving the thread structure and original user ids and finally iii) allows the user to a) label the comments with concepts, b) group the concepts c) create summary and finally d) back link the summary sentences to the groups created in b.[3]

[3] The tools can be accessed through 143.167.10.89/senseiAnnotation.



Stage 2 - Grouping

For reference you may:
Click to view the original comments and your labels in a new tab

Please sort your labels into groups, creating a "group label" when you have identified common content for a group.

Remember to save your work regularly:

When you have finished grouping your labels please:

Use the buttons to add grouping tags:

These are the labels you have provided for the comments:

```
GROUP: NR do not set fares / operate trains

label for comment 2 [Cynic24]: NR do not set fares / operate trains
label for comment 3 [Cynic24]: NR do not set fares / operate trains
label for comment 4 [lindalusardi]: NR do not set fares - accepted
label for comment 6 [Craig Axon]: Ticket prices set by government; NR not-for-profit; system not transparent
label for comment 8 [PrimitivePerson]: NR do not set fares
label for comment 9 [Cynic24]: NR do not set fares. TOCs set fares, under government restrictions.
label for comment 11 [C2H4n]: TOCs / NR separate.
label for comment 12 [Cynic24]: NR do not set fares / operate trains
label for comment 18 [Cynic24]: NR do not set fares

GROUP: NR is non-profit / "not for dividend"

label for comment 6 [Craig Axon]: Ticket prices set by government; NR not-for-profit; system not transparent
label for comment 34 [Cynic24]: NR non-profit. Income ploughed back, or to government.
label for comment 35 [C2H4n]: NR "not for dividend" private company.
label for comment 36 [Cynic24]: NR "not for dividend" private company - but becoming public body soon.

GROUP: fares too high
```

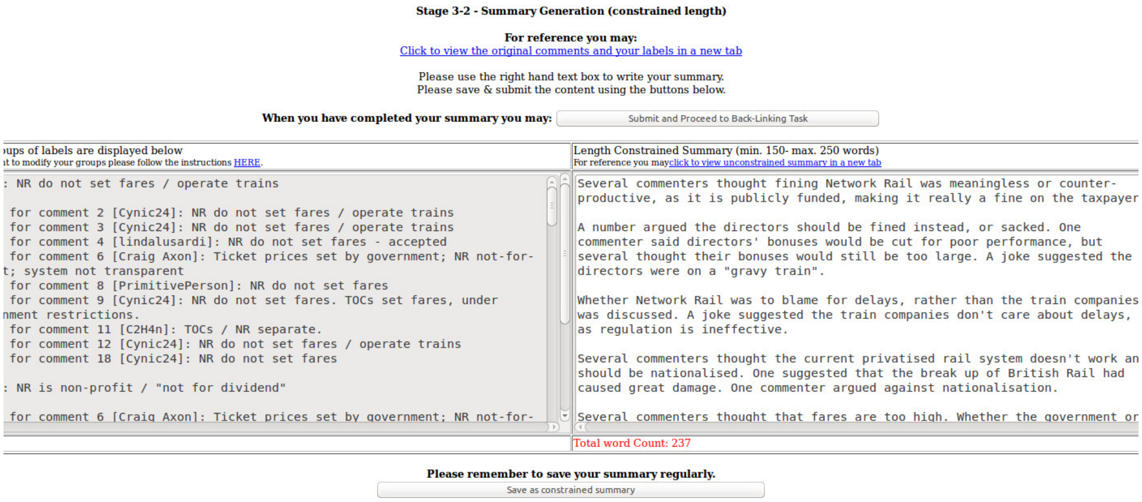
Figure 7: Window for Labelling Grouping.

For labeling (tast a) for each comment the interface provides a cell to hold a label annotation for that comment. Different labels are separated by a semicolon. Figure 6 shows a screen shot of the window for labeling comments. In the next task the annotators can group the labels (shown in Figure 7). In this stage annotators can create their own group labels and copy past all the comment labels under the created group label. Note we main groups to have sub-groups. However, sub-groups cannot have further sub-groups. Once the grouping is finished the annotator is able to generate summaries. As described in the previous section we distinguish between unconstrained and constrained summaries . In the unconstrained case, written first, the annotator create a summary without any lenght limit; in the constrained one the summary length must be between 150 and 250 words. The interface supporting summary writing for the constrained summary is shown in Figure 8 (the interface for the unconstrained summary is essentially the same, but does not enforce a length limitation).

Finally, once the summaries have been saved, we generate a version of the summary split into its component sentences. The annotator can then back link the summary sentences to the group/sub-group labels (shown in Figure 9). Annotators first select a sentence in the summary by clicking on it (it should be shown in a green colour highlight). Then you check-mark any relevant groups from the set shown below on the screen. When you have made your selection you should save the links and proceed to select a new sentence. If you change your mind you may go back and add/remove previous saved links, but remember to save the changes.

Note our tool supports English and French.[4] The tool itself is written language indepently (the displayed sentences/information in the interface are not hard-coded). This means the tool can display any language of choice. For this we maintain a property file containing key-values. The key elements are used

[4] In the start of the tool the annotator can decide what language to work on. According to the language also the correct annotation data will be uploaded.



Stage 3-2 - Summary Generation (constrained length)

For reference you may:
Click to view the original comments and your labels in a new tab

Please use the right hand text box to write your summary.
Please save & submit the content using the buttons below.

When you have completed your summary you may:

ıups of labels are displayed below
ıt to modify your groups please follow the instructions HERE.

```
: NR do not set fares / operate trains

 for comment 2 [Cynic24]: NR do not set fares / operate trains
 for comment 3 [Cynic24]: NR do not set fares / operate trains
 for comment 4 [lindalusardi]: NR do not set fares - accepted
 for comment 6 [Craig Axon]: Ticket prices set by government; NR not-for-
t; system not transparent
 for comment 8 [PrimitivePerson]: NR do not set fares
 for comment 9 [Cynic24]: NR do not set fares. TOCs set fares, under
nment restrictions.
 for comment 11 [C2H4n]: TOCs / NR separate.
 for comment 12 [Cynic24]: NR do not set fares / operate trains
 for comment 18 [Cynic24]: NR do not set fares

: NR is non-profit / "not for dividend"

 for comment 6 [Craig Axon]: Ticket prices set by government; NR not-for-
```

Length Constrained Summary (min. 150- max. 250 words)
For reference you mayclick to view unconstrained summary in a new tab

```
Several commenters thought fining Network Rail was meaningless or counter-
productive, as it is publicly funded, making it really a fine on the taxpayer

A number argued the directors should be fined instead, or sacked. One
commenter said directors' bonuses would be cut for poor performance, but
several thought their bonuses would still be too large. A joke suggested the
directors were on a "gravy train".

Whether Network Rail was to blame for delays, rather than the train companies
was discussed. A joke suggested the train companies don't care about delays,
as regulation is ineffective.

Several commenters thought the current privatised rail system doesn't work an
should be nationalised. One suggested that the break up of British Rail had
caused great damage. One commenter argued against nationalisation.

Several commenters thought that fares are too high. Whether the government or
```

Total word Count: 237

Please remember to save your summary regularly.

Figure 8: Window for Constrained Summary Authoring.

within the interface to select the values. The values can be adopted to any language of choice. So far we have created values for English and French. The creation of values means to translate an existing value (usually in English) to e.g. French. The translation of all values does not take more than 1 hour.

4.1.4 Annotators and training

We recruited graduates with expertise in writing or language related studies and final year journalism students. The majority of annotators were native English speakers, the others had excellent English writing skills. We also recruited 2 Italian annotators to carry out Italian text summary writing.

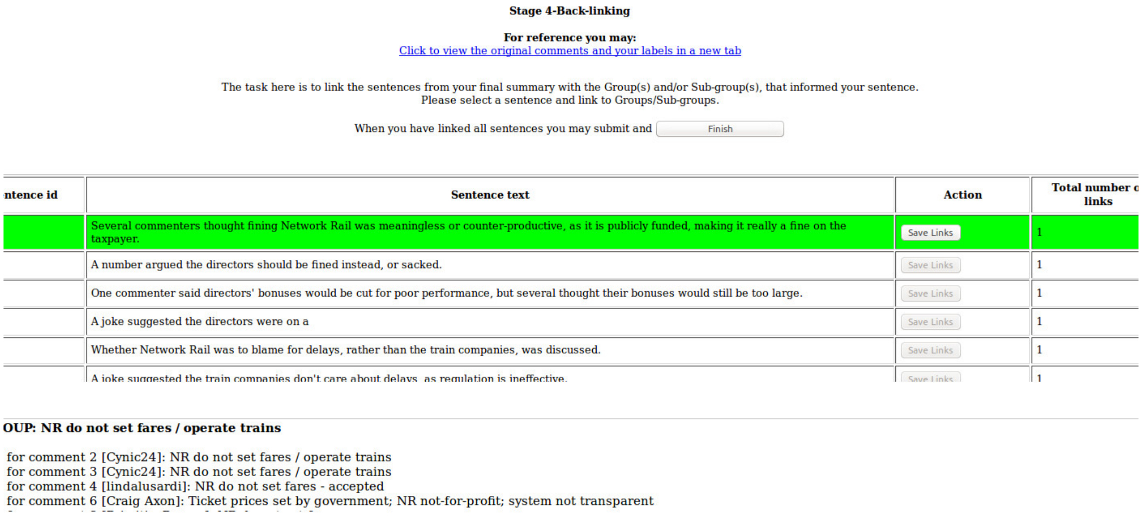
Each annotator carried out a training session taking 1.5-2 hours.

4.1.5 Corpus Design

In this initial phase of Gold Standard development we planned for a full set of annotations for each of 20 article and comment sets (in English) in a variety of topics. A full set of annotations = Constrained/unconstrained length summaries, a set of labeled comments, Groups of labels and comments, (each label pointing to the underlying comment), Backlinked summaries to groups (providing a record of the groups of labels and comment that informed a summary sentence).

We planned for each comment set to be double annotated, but allowed for further annotations if time was available.

In addition we planned for 2 Italian articles to be annotated, each by two annotators.



Stage 4-Back-linking

For reference you may:
Click to view the original comments and your labels in a new tab

The task here is to link the sentences from your final summary with the Group(s) and/or Sub-group(s), that informed your sentence.
Please select a sentence and link to Groups/Sub-groups.

When you have linked all sentences you may submit and

ntence id	Sentence text	Action	Total number c links
	Several commenters thought fining Network Rail was meaningless or counter-productive, as it is publicly funded, making it really a fine on the taxpayer.	Save Links	1
	A number argued the directors should be fined instead, or sacked.	Save Links	1
	One commenter said directors' bonuses would be cut for poor performance, but several thought their bonuses would still be too large.	Save Links	1
	A joke suggested the directors were on a	Save Links	1
	Whether Network Rail was to blame for delays, rather than the train companies, was discussed.	Save Links	1
	A joke suggested the train companies don't care about delays, as regulation is ineffective.	Save Links	1

OUP: NR do not set fares / operate trains

for comment 2 [Cynic24]: NR do not set fares / operate trains
for comment 3 [Cynic24]: NR do not set fares / operate trains
for comment 4 [lindalusardi]: NR do not set fares - accepted
for comment 6 [Craig Axon]: Ticket prices set by government; NR not-for-profit; system not transparent

Figure 9: Window for Backlinking Summary Sentences to Clusters of Comment Labels.

4.1.6 Source Data

Training data We downloaded 3,362 news articles along with their comments from the Guardian.com news paper over a period of two months (June-July 2014) . Guardian.com provides for each topic e.g. business, politics, etc. a specific RSS feed URL. We manually collected RSS feeds for the topics: politics, health, education, business, society, media, science, the-northener, law, world-news, scotland-news, money and environment. Using an in-house tool we visited the news published through the RSS feeds every 30 minutes, downloaded the article content and also recorded the news URL. Every recorded news URL was re-visited after a week (the time we found sufficient for an article to attract commenters) to obtain its comments. Articles contained between 1 and 6,223 comments, averaging 425.95 (median 231.5) comments per article.

Testing data From this collection of articles we selected 25 news articles for annotation. The annotated corpus to date incudes 18 news articles and their associated comments.

The topics covered by the 18 articles are politics, sport, health, environment, business, scotland-news and science. The articles have min. 406 words, max. 2084 words and on average 746 words. The size of the comment sets for the 18 articles varies: the minimum number of comments is 100, maximum 1076 and the average is 372. The average number of words in a comment is 52; (min 1 word and max 839 words).

For annotation purposes we extracted the first 100 comments from the full comment stream[5]. The average number of comments in the annotation set is 103, (min. 100, max. 116 comments). However, the comment sets for annotation vary in terms of total number of words and average number of words

[5] Note: Threads were ordered chronologically (i.e. oldest first). If the thread containing the 100^{th} comment had further comments we continued including those comments until the last comment in that thread.

per comment: the average total word count in an annotation comment set was 4910.8, and the average number of words in a comment, calculated over all 18 comment sets is 48.1, but the smallest comment set, totalled just 3141 words with an average number of words per comment of just 23.66, while the largest comment set had a total word count of 8739, with an average of 86.52 words in a comment.

4.1.7 Gold standard Annotations

18 articles and associated comment sets (at least 100 comments per comment set, as described in the last section) were double annotated and 2 of the 18 have been triple annotated. So, 18 articles have 2 full sets of annotations and 2 of the 18 articles have 3 full annotation sets. In total there are 38 English summaries; all are coherent and fluent summaries. Each summary includes multiple examples of the key ingredients of a summary according to our definition in section 2.2 above; i.e., they identify issues and characterise opinion on the issues. As part of the process of creating each summary (see Section 4.1.2 above), annotators have grouped comment labels which, since comment labels are uniquely associated with comments, allows us to assemble groups or clusters of topically related comments. We have done this to obtain gold standard clusters for evaluating our clustering algorithms. On average there are 8.97 human-created clusters per comment set in the gold standard cluster data set.

4.2 Comment-Article Linking

User comments on news articles and other online content provide a communication channel between journalists and their audience, which has replaced the previously prevalent one-way reporting from journalists to their readers. Online commenting is therefore a feature which several user groups in media business crucially depend on in their attempts to build and maintain their reputation and a wide readers and customer base. To achieve this, however, it is essential to foster high quality discussions in online commenting forums because quality and tone of comments are shown to influence the readers' attitudes to online news content [6, 7, 8].

In the present setup of online forums, comments are difficult to organize, read and engage with, which affects the quality of discussion and the usefulness of comments for the interested parties. One problem with comments in their current form is their detachment from the original article. Placed at the end of the article without clear reference to the parts of the article that triggered them, comments are hard to put into the context from which they originated, and this makes them difficult to make sense of and evaluate. Comment-article linking is also necessary in more complex systems for information extraction from comments such as comment summarization [9, 10, 11, 12, 13]. Such systems rely on identifying relevant comments and those that link to the articles are good candidates.

In this section we report the results of our experiments in comment-article linking. Specifically, the task is to bring together readers' comments with the online news article segments that the comments refer to. We perform two different evaluations: in-domain and out-domain. In the in-domain evaluation we report the results of our approach obtained through the comment-article linking shared task organized by our partners at the University of Essex and reported in Section 3.1.1 of D7.4. In the out-domain evaluation we use data from related work and compare the performance of our approach to that of more elaborate topic modelling methods such as the ones proposed by Sil et al. [14] and Das et al. [15] and demonstrate that comparable linking results can be achieved by simpler text similarity methods.

We start with defining the linking task and the pre-processing steps we perform on the article and comments (Sections 4.2.1 and 4.2.2). Then we provide the description of our linking approach (Section 4.2.3). In Sections 4.2.4 and 4.2.5 we report our experimental results.

4.2.1 The task

For the linking task we assume a news article A is divided into n segments $S(A) = s_1, ..., s_n$. The article A is also associated with a set of comments $C(A) = c_1, ..., c_l$. The task is to link comments $c \in C(A)$ with article segments $s \in S(A)$. We express the strength of link between a comment c and an article segment s as their linking score ($Score$). A comment c and an article segment s are linked if and only if their $Score$ exceeds a threshold, which we experimentally vary. $Score$ has the range [0,1], 0 indicating no linking and 1 defining a strong link.

4.2.2 Pre-processing

First, we split the news article into segments. To allow for the comparability of the results, we comply with segmentation decisions used in previous work ([14], [15]) and treat each article sentence as a segment. Each comment is treated as one unit regardless how many sentences it has. Then each sentence-comment pair is pre-processed before it is analyzed for linking. The example below illustrates the outputs of the pre-processing pipeline.

The pre-processing includes tokenization[6] and lemmatization (shown in (2) in the example below, where an original article sentence is shown in (1)). Next, we use either words with stop-word removal (shown in (3)) or terms (shown in (4) where each term is split by a semicolon) to represent the article sentence and also each comment. Terms are extracted using the freely available term extraction tool TWSC[7] [16]. We also record named entites (NEs) (shown in (5)) extracted from the article segment (comment).

1. **Original article sentence:** *An Afghan policewoman walked into a high-security compound in Kabul Monday and killed an American contractor with a single bullet to the chest, the first such shooting by a woman in a spate of insider attacks by Afghans against their foreign allies.*

2. **After tokenization and lemmatization:** *an afghan policewoman walk into a high - security compound in kabul monday and kill an american contractor with a single bullet to the chest , the first such shooting by a woman in a spate of insider attack by afghan against their foreign allies .*

3. **When words are used:** *afghan, policewoman, walk, high, security, compound, kabul, monday, kill, american, contractor, single, bullet, chest, shooting, woman, spate, insider, attack, afghan, foreign, allies*

[6] For shallow analysis we use the OpenNLP tools: https://opennlp.apache.org.

[7] TWSC uses POS-tag grammars to recognize terms. This leads to cases that it recognises base NP-like word sequences that we refer to as terms. Terms are extracted from the original version of the sentences but words in the terms are replaced with their lemmas.

4. **When terms are used:** *shooting by a woman;woman in a spate; spate of insider; compound in kabul; kabul monday; insider attack; afghan policewoman; american contractor; single bullet; security compound; foreign allies; policewoman; security; compound; contractor; bullet; chest; shooting; woman; spate; insider; attack; allies; afghan; kabul; monday*

5. **Extracted NEs:** *Kabul*

4.2.3 Algorithm

We investigate a method of linking comments and news article sentences using a linear combination of similarity scores as computed through a number of different similarity metrics (features). However, some comments quote article segments, therefore explicitly linking comments to article segments. To account for this, we consider a comment and an article sentence linked if their quotation score ($quoteScore$) exceeds a threshold. Otherwise, a similarity score is computed and articles are linked if their similarity score is above a threshold. The following paragraphs describe how features and thresholds are computed.

Each metric is computed based on the comment $c \in C(A)$ and a segment $s \in S(A)$ as input. We pair every segment from $S(A)$ with every comment from $C(A)$. With this set up we are able to link one to many comments with one segment and also one to many segments with a particular comment, which implements the *n* to *m* comment-segment linking.

Quotation Based Linking We link all comments including quotes to the article sentences they quote. To determine whether a segment is quoted in the comment we compute $quoteScore = \text{len}(quote)/\text{len}(S)$ with *len* [8]. *len* returns the number of words of the given input and *quote* is a place holder for consecutive news article words found in the same order within the comment. If the $quoteScore$ exceeds an experimentally set threshold of 0.5 (50% of consecutive article segment words are found in the same order within the comment), then the segment is regarded as quoted in the comment, the comment-segment pair is linked, their linking $Score$ is set to $quoteScore$ and no further linking features are considered. However, qualitative observations on random data portions have shown that only sentences longer than 10 words render meaningful quote scores, so we add this as an additional constraint.

Similarity Linking

Similarity Feature Extraction If a comment does not contain a quote as described above, we compute the following features to obtain the value of the similarity score without considering the quote feature:

- **Cosine:** The cosine similarity [17] computes the cosine angle between two vectors. We fill the vectors with terms/word frequencies extracted from the article segment/comment.

[8] For this feature the orginal version, i.e. without pre-processing, of article segment and comment are used.

- **Dice:**
$$dice = \frac{2 * \text{len}(I(S,C))}{\text{len}(S) + \text{len}(C)} \quad (1)$$
where $I(S,C)$ is the intersection set between the terms/words in the segment and in the comment. *len* returns the number of entries in the given set.

- **Jaccard:**
$$jaccard = \frac{\text{len}(I(S,C))}{\text{len}(U(S,C))} \quad (2)$$
where $U(S,C)$ is the union set between the terms/words in the segment and comment.

- **NE overlap:**
$$NE_{overlap} = \frac{\text{len}(I(S,C))}{\text{len}(U(S,C))} \quad (3)$$
where $I(S,C)$ is the intersection set between the named entities (NEs) in the segment and in the comment and where $U(S,C)$ is the union set between the NEs.

- **DISCO 1 + DISCO 2:** *DISCO* (DIStributionally similar words using CO-occurrences) assumes words with similar meaning occur in similar context [18]. Using large text collections such as the BNC corpora or Wikipedia, distributional similarity between words is computed by using a simple context window of size 3 words for counting co-occurrences. DISCO computes two different similarities between words: *DISCO1* and *DISCO2*. In DISCO1 when two words are directly compared for exact similarity DISCO simply retrieves their word vectors from the large text collections and computes the similarity according to Lin‘s information theoretic measure [19]. DISCO2 compares words based on their sets of distributional similar words. Note, we use the Wikipedia corpus for computing both (DISCO 1 and 2) distributional similarities.

Computing Similarity Linking Score Using a linear function we combine the scores of each of these features (*cosine* to *DISCO*) to produce a final similarity score for a comment-segment pair:

$$Score = \sum_{i=1}^{n} feature_i * weight_i \quad (4)$$

where $weight_i$ is the weight associated with the i^{th} feature. The weights are trained based on linear regression using the Weka package and the training data described in the following section.

Training Data Obtaining training data requires manual effort and human involvement and is thus very expensive, while resulting in relatively small training data sets. We therefore automatically assemble training data by using comments with article quotes as a training data set. As outlined above, in addition to original comment text, many comments include a brief quotation from the article, therefore directly indicating which article segments have triggered the comments. The set of comments with quotes linked to the article segments they quote are therefore used as our training data.

As our training data we use the news articles described in Section 4.1.6. Each article in this data set was split into sentences and for each of these sentences (containing at least 10 words) it was determined whether it is quoted in any of the comments as described above. In case the $quoteScore$ was above 0.5 for a sentence-comment pair, the pair was included in the training set. Using this process we have extracted 43,300 sentence-comment pairs to use for training. For each pair the similarity features listed in Section 4.2.3 were extracted. The $quoteScore$ was used as the expected outcome. We also included 43,300 negative samples into the training data in order to present linear regression with the behaviour of the features for wrong sentence-comment links. The negative samples were created by pairing every sentence containing at least 10 words of article X with every comment of article Y. In this way we pair comments with sentences of another article that have not originally triggered the comments. Similar to the positive samples, the quote score was taken as the expected outcome. However, unlike the positive samples, the $quoteScore$ threshold of 0.5 was not applied for the negative samples.

4.2.4 In-domain evaluation

The in-domain evaluation is performed within the comment-article linking shared task organized by our partners the University of Essex. Similar to the task described in Section 4.2.1 the linking task within the shared task was to link a comment to an article segment (sentence). However, unlike the task described above the comment was not treated as one unit but split into sentences. This allowed to link parts of the comment (sentences) to article sentences and leave some out. Although the shared task set-up defined this freedom within the comments USFD continued treating the entire comment as one unit. More precisely, when our linking approach found a link between a sentence in the comment and an article sentence it also linked all the remaining sentences within the comment to the article sentence. The evaluation was performed with English and Italian data. For more details about the data please see the report by the University of Essex.

Each participant was allowed to submit two runs. Our runs differed in how we set a threshold for linking similarity. The first run was set to a lower threshold (i.e. the $Score$ in equation 5 was set to 0.3). Anything below this threshold was not linked. In the second run the threshold was set to 0.5. For English both our runs were considered. However, for Italian there has been some problems in the submission, so that our second run with the threshold 0.5 was not considered.

Our results for English are that using our second run we obtained better results compared to all other 8 system submissions. With this set-up we achieved 89% precision. Our first run (run with the 0.3 threshold) achieved 82% precision. With this score it became the 5th system. For Italian our first run got the 6th position scoring 89% precision. Since our first run also did not perform well on the English data, it is likely that the performance on the Italian data would have been also better with the second run.

4.2.5 Out-domain evaluation

Test Data In the out-domain evaluation we use the AT data to test our linking method. The AT data set is reported in [15] and consists of articles with comments downloaded from the technology news website *Ars Technica* (AT). In this data set there are 501 articles. Each article contains min. 8, max. 132 and avg. 38 sentences. Each article is assigned min. 2, max. 59 and avg. 6.3 comments. As reported in [15] two annotators mapped comments to article sentences; however, the agreement between annotators

Table 7: Comparison of term/word based similarity metrics on article-comment linking to SCTM and Corr-LDA.

Method	Precision	Recall	F1
Metrics$_{term}$	**0.512**	0.292	0.372
Metrics$_{word}$	0.316	0.30	0.31
Metrics$_{termWord}$	0.414	0.31	0.356
SCTM	0.36	0.44	0.39
Corr-LDA	0.01	0.03	0.01

cannot be assessed from the available data set due to the lack of double annotations. Below we show an example article sentence (6) and a comment (7), which was identified as linked to the sentence in (6) by one of the annotators.

1. **Article Sentence:** The capacity decreases over 30 charging cycles, meaning that the battery held less energy each time it was recharged.
2. **Comment:** Decreases over 30 charges? So one part of the battery's made with more common ingredients making it sorta-partially-almost cheaper, but you're only good for 30 charges? I'm sure it's been stated in this thread already, but that number itself seems counterproductive.

State-of-the art The combined quotation and similarity based linking investigated here is compared to the state-of-the-art SCTM method reported in [15]. SCTM (Specific Correspondence Topic Model that admits multiple topic vectors per article-comment pair) is an LDA-based topic modeling method, which has been developed by Das et al. [15] in order to account for the multiplicity of topics in comments and articles. Their baseline is *Corr-LDA*, which Das et al [15] deem unsuitable since it is restricted to using only a single topic vector per article-comment pair. Evaluation on the same AT test data set allows for a direct comparison of our results to those of SCTM and Corr-LDA. Another recently proposed linking approach is reported in [14]. However, it does not match the performance of its simple $tf * idf$ based baseline, so we do not consider this method in our evaluations.

Results Table 7 shows the performance of the automated linking using quotation and similarity metrics ($Metrics$) on the AT data. The table shows the results for both term and word based representation of article segments (first two rows). Both results were obtained with the experimentally determined $Score >= 0.5$. The results in the table show that representation of article segments and comment texts as terms is superior to the bag-of-words representation for the comment-article linking task as it achieves substantially higher score in precision with a similar recall value. We also combined terms with words by merging the term list with the bag of words and used them to compute the metrics. The results are shown in the 3^{rd} row. Compared to the word only variant, $Metrics_{word}$, we see a substantial improvement in the precision and a slight one in the recall score. However, compared to the term only variant, $Metrics_{term}$, the precision score is still low indicating that terms only are indeed the better choice for representing article segments and comments for the linking task.

The results in Table 7 show that the state-of-the-art baseline SCTM outperforms the $Metrics$ regarding the overall F1 score due to higher recall. However, this difference in F1 score is small. The precision of $Metrics_{term}$ based similarity is substantially higher than that of the SCTM method at the expense of recall. Higher precision may be preferable to higher recall for the linking task as including wrong links in order to have higher coverage is more noisy and therefore more disturbing for both human and automatic processing of comment-article links than leaving relevant comments unlinked. These results suggest that term based similarity linking is performing almost as well as the SCTM method overall, and if increasing precision over recall is favored for the comment-article linking task, it even could be a preferred method for this task.

4.3 Comment Clustering

Online news outlets attract large volumes of comments every day. For Huffington Post for example, it has been estimated that it receives around 140,000 comments in 3 days [9], while Guardian has been reported to have 25000 to 40000 comments per day[10]. These figures suggest that online commenting forums are important not only for users to share their opinions on relevant recent news, but also for multiple stakeholders in media business. All user groups involved in online commenting to news would profit from an easier access to multiple topics discussed within a large set of comments. For example, commenters would be able to gain a quick overview of already discussed topics and insert their contributions at a relevant place in the discussion. Journalists who wrote the news article would have an access to multiple conversation topics that their article has triggered and would be able to engage with their readers in a more focused way. Editors would be able to monitor the topics that are most interesting to the readers, comment forum moderators' work would be easier and marketers could use conversations grouped around topics for developing personalized marketing strategies.

In the current set up of commenting forums the comments are grouped into threads - micro-conversations within the large set of comments to an article that are set up and expanded by user themselves, occasionally with some intervention of moderators. As all freely developing conversations, threads tend to digress in their topic structure and contain multiple topics. Comments addressing a particular conversation topic can occur in multiple threads or as stand-alone comments. Therefore, at present there is no easy way to group comments into topics in online comment forums to news.

Grouping readers' comments to online news articles according to topics they address presents several challenges. For example: (1) the number of topics discussed in a conversation about a news article is always unknown. (2) reader comments can be multi-topical themselves, therefore, one comment can belong to different topic clusters. (3) Comments that implement a conversational action like jokes, sarcastic remarks or short support items (e.g. 'Great') are typically difficult to assign to a topic cluster according to their content.

We investigate graph-based clustering for addressing the first two challenges. Graph-based clustering has been applied for various tasks such as image segmentation [20], clustering medical data [21, 22], text summarization [23], documents clustering [24], etc. A detailed review of graph clustering can be found in Schaeffer [25]. The work on clustering news comments on the other hand is sparse at present. A recent study by Llewellyn et al. [26] applies LDA, K-Means as well as simple metrics such as cosine

[9] http://goo.gl/3f8Hqu

[10] http://www.theguardian.com/commentisfree/2014/aug/10/readers-editor-online-abuse-women-issues

measure and clustering by key words to cluster news comments. They have demonstrated superiority of LDA [27] above the other approaches for this task based on clustering comments of only one news article. Therefore we use LDA as our baseline, despite its obvious limitation in requiring the number of possible topics in advance.

4.3.1 Algorithm

Our graph-based clustering approach is based on the Markov Cluster Algorithm (MCL) [28] shown in Algorithm 1. The nodes (V) in the graph $G(V,E,W)$ are the comments. Edges (E) are created between the nodes V and have associated weights (W). Each comment is potentially connected to every other comment using an undirected edge. The edge is present if the associated weight is greater than 0. We also allow nodes to link to themselves. The edge weights are computed based on comment-comment similarity features described below. Once such a graph is constructed, MCL repeats steps 6-8 in the Algorithm until the maximum number of iterations $iter$ is reached[11]. First in step 6 the matrix is normalized, next expanded (step 7) and finally inflated (step 8). 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. These two operations are controlled by two parameters, the power *p* and the inflation parameter *r*. After some experimentation we set both parameters to *2*, as this resulted in a good balance between too many and too few clusters.

Algorithm 1 MCL Algorithm

```
Require: un-directed graph $G(V,E,W)$, power parameter $p$, inflation parameter $r$, matrix $M_{mn}$ with $m = n = |V|$,
    number of iterations $iter$
 1: for all $e_{ij}$ do
 2:     Fill matrix cells with edge weights ($M_{ij} = w_{ij}$)
 3: end for
 4: Add self loops to each node ($M_{ii} = 1$)
 5: repeat
 6:     Normalize $M$ ($M_{ij} = M_{ij} / \sum_{j=1}^{n} M_{ij}$)
 7:     Expansion: Raise $M$ to the $p^{th}$ power
 8:     Inflation: $M_{ij} = (M_{ij})^r$
 9: until current iteration $\leq iter$
10: Extract clusters from the final matrix
```

Once MCL terminates, the clusters are read off the rows of the final matrix (step 10 in Algorithm 1). For each row i in the matrix the comments in columns j are added to cluster i if the cell value $M_{i,j} > 0$ (the rows for items that belong to the same cluster will each redundantly specify that cluster). In this setting the MCL algorithm performs hard clustering, i.e. assigns each comment to exactly one cluster.

Weighting edges between comments To weight an edge between two comments C_1 and C_2 we use the Cosine, Dice, Jaccard, NE overlap features described in Section 4.2.3. In addition to these we also use:

[11] MCL runs a predefined number of iterations and stop after reaching them. We run MCL with 5000 iterations.

- **Same thread**: If both C_1 and C_2 are within the same thread this feature returns 1 otherwise 0.
- **Reply relationship**: If C_1 replies to C_2 (or wise versa) this feature returns 1 otherwise 0. Note this reply relationship does not need to be direct, i.e. a comment C_x can reply to comment C_y that replies to C_z. In this case we also treat C_x as replying to C_z.

We use weighted linear combination of these features and train weights using linear regression where the target value for positive instances is $quoteScore$ as defined below and for negative examples is 0:

$$Score = \sum_{i=1}^{n} feature_i * weight_i \tag{5}$$

We create an edge within the graph if the $Score$ is above 0.3, a threshold value set experimentally. The value of $Score$ is used to weight the edges. The feature weights are trained using training data derived from the set of news articles and comments described in Section 4.1.6.

We build positive and negative comment-comment pairs from *The Guardian* data. To construct positive pairs we assume that if two or more comments associated with the same news article quote the same sentence in that article, then they are on the same topic and thus belong to the same topic cluster, i.e., positive pairs consist of comments that quote the same article sentence. When computing comment-comment similarity, if quotes are left in the comments, the similarity metric may be biased by the exact match as found in the quotes and may not be sensitive enough to capture similarity in comments that do not contain exactly matching quotes. For this reason, we expect that clustering results will be better if quotes are removed from the comments before computing similarity. To test this assumption we created two sets of training data. In the first set we have positive paired comments where the quote is still in the comments. The second training data set contains positive pairs of comments where we removed the shared quotes from the comments. Note, in both training sets we use for each pair the $quoteScore = len(quote_{C1}) + len(quote_{C2})/2 * len(sentence)$ as the outcome. $len(X)$ returns the length of X in words and $quote_{Ci}$ is the segment of Ci quoted from $sentence$ in the original article. When computing the $quoteScore$ we make sure that the quoted sentence has at least 10 words. We add comment pairs to the positive training data whose $quoteScore$ values are $>= 0.5$ – a value we obtained empirically.

The negative instances were created by pairing randomly selected comments from two different articles from *The Guardian*. They are used to present the linear regression algorithm with the instances of comment pairs that are not on the same topic or are only weakly topically related. We have in total 14,700 positive pairs and the same number of negative instances. The outcome of each such pair was set to 0.

4.3.2 Evaluation

In the cluster evaluation we use the gold standard clusters created by the USFD team. The data is described in Section 4.1.7

LDA We compare our graph based approach against LDA. LDA is a well known topic modelling approach. Llewellyn et al. [26] applied several approaches LDA [27] to group similar comments. The authors have shown that clustering news comments using LDA performs best compared to the other systems such as K-means and simple distance metrics. For this reason we compare our results with those of LDA.

We use two different LDA models: *LDA1* and *LDA2*. The LDA1[12] model is trained on the entire training data set described in Section 4.1.6. In this model we have treated the news article and its comments as a single document. This training data set is large and contains a variety of topics. When we require the clustering method to identify a small number of topics, we expect these to be very general, so that the resulting comment clusters are less homogeneous, than they would be if only comments of a single article are considered as it is the case in Llewellyn et al. [26].

Therefore we also train a second LDA model (LDA2), which replicates the setting reported in Llewellyn et al. [26]. The LDA2 model is trained on the comments of the testing articles described in Section 4.1.6. For each article we include the entire comment set in the training data, which contains the first 100 comments that are annotated by human annotators, as well as the remaining comments that are not included in the gold standard. In obtaining LDA2 we treated each comment in the set as separate document.

LDA requires a predetermined number of topics. We set the number of topics to 9 since the average number of clusters within the gold standard data is 8.97. We use 9 topics within both LDA1 and LDA2. Similar to Llewellyn et al. [26] we also set the α and β parameters to 5 and 0.01 respectively for both models.

Once the models are generated they are applied to the test comments for which we have gold standard clusters. LDA distributes the comments over the pre-determined number of topics using probability scores. Each topic score is the probability that the given comment was generated by that topic. Like [26] we select the most probable topic/cluster for each comment. Implemented in this way, the LDA model performs hard clustering.

Results For evaluation the automatic clusters are compared to gold standard clusters described in Section 4.1.7. Amigo et al. [29] report several metrics to evaluate automatic clusters against the gold standard data. However, they are tailored for hard clustering. Although our graph-based approach and baseline LDA models perform hard clustering, the gold standard data contains soft clusters. Therefore, the evaluation metric needs to be suitable for soft-clustering. In this setting hard clusters are regarded as a special case of possible soft clusters and will likely be punished by the soft-clustering evaluation method. We use fuzzy BCubed Precision, Recall and F-Measure metrics reported in [30] and [31]. According to the analysis of formal constraints that a cluster evaluation metric needs to fulfill [29], fuzzy BCubed metrics are superior to 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 fuzzy metrics are also applicable to hard clustering.

Clustering results are shown in Table 8. A two-tailed paired t-test was performed for a pairwise comparison of the fuzzy Bcubed metrics across all four automatic systems and human-to-human setting.

[12] We use the LDA implementation from `http://jgibblda.sourceforge.net/`.

Table 8: Cluster evaluation results. The scores shown are macro averaged. For all systems the metrics are computed relative to the average scores over Human1 and Human2. *graphHuman* indicates the setting where similarity model for graph-based approach is trained with quotes included in the comments

Metric	Human1-Human2	graph-Human	graph-Human-quotesRemoved	LDA1-Human	LDA2-Human
Fuzzy B^3Precision	0.41	0.29	0.30	0.25	0.23
Fuzzy B^3Recall	0.44	0.30	0.33	0.29	0.17
Fuzzy B^3FMeasure	0.40	0.29	0.31	0.24	0.18

Firstly, we observe that human-to-human clusters are significantly better than each of the automatic approaches in all evaluation metrics[13]. Furthermore, we cannot retain our hypothesis that the graph-based approach trained on the training data with quotes removed performs better than the one that is trained on data with quotes intact. Although the results in the *quotes removed* condition are better for all metrics, none of the differences is statistically significant. We use the better performing model (graph without quotes) for comparisons with other automatic methods.

Secondly, the LDA1 baseline performs significantly better than the re-implementation of previous work, LDA2, in all metrics. This indicates that training LDA model on the larger data set is superior to training it on a small set of articles and their comments, despite the generality of topics that arises from the compressing topics from all articles into 9 topic clusters for LDA1.

Finally, the *quotes removed* graph-based approach (column 4 in Table 8) significantly outperforms the better performing LDA1 baseline in all metrics. This indicates that graph-based method is superior to LDA, which has been identified as best performing method [26]. In addition, clustering comments using graph-based methods removes the need for prior knowledge about the number of topics - a property of the news comment domain that cannot be considered by LDA topic modelling.

4.3.3 Discussion

The comment clustering results demonstrate that graph-based clustering is more suitable for clustering reader's comments to online news into topics than the current state-of-the-art method LDA as implemented in previous work.

We also performed a qualitative analysis on the results of the graph-based clustering approach. The analysis reveals that disagreements in human and automatic assignment of comments to clusters are frequently due to the the current approach favouring a flat conversational structure, where each comment is regarded as independent. Commenting forums, however, are conversations and as such they exhibit internal structuring where two comments are functionally related to each other, so that the first pair part (FPP) makes relevant the second pair part (SPP). In our automatic clusters we frequently found answers, questions, responses to compliments and other stand-alone FPPs or SPPs that were unrelated to the rest of otherwise homogeneous cluster. For example, the comment *'No, just describing another right wing asshole.'* is found as the only odd comment in otherwise homogeneous cluster of

[13] The difference in these results is significant at the Bonferroni corrected level of significance of $p < 0.0125$, adjusted for 4-way comparison between the human-to-human and all automatic conditions.

comments about journalistic standards in political reporting. Its FPP 'Wait, are you describing Hillary Clinton?' is assigned to a different cluster about US politician careers. We assume that our feature *reply structure* was not sufficiently weighted to account for this, so that we need to consider alternative ways of training, which can help identify conversational functional pairs.

A further source of clustering disagreements is the fact that humans cluster both according to content and to the conversational action a comment performs, while the current system only clusters according to a comment's content. Therefore, humans have clusters labelled '*jokes*', '*personal attacks to commentors or empty sarcasm*', '*support*', etc., in addition to the clusters with content labels. A few comments have been clustered by the annotators on both dimensions, content and action, and can be found in both clusters (soft clustering). Our graph-based method reported in this work produces hard clusters and is as such comparable with the relevant previous work. However, we have not addressed the soft-clustering requirement of the domain and gold standard data, which has most likely been partly reflected in the difference between human and automatic clustering results. When implementing soft-clustering in future one way to proceed would be to add automatic recognition of a comment's conversational action, which would make graph based clustering more human-like and therefore more directly comparable to the gold standard data we have.

4.4 Cluster Labelling

The previous section gave details about comment clustering. Even though comments are clustered, they have not been labeled yet. Without labels, it is difficult to have a comprehensive picture of what the comments represent. Assigning labels is not an easy task as a cluster of comments can have multiple dimensions representing various topics of analysis. Moreover, the granularity at which these labels can be assigned differs. They can be a generic name or they can be very specific, adapted to the domain of the topic. For instances, if the topic words selected from a comment cluster are *rate, population, prevalence, study, incidence, datum, increase, mortality, age and death* and possible labels can be as generic as *population study* or be very specific like *mortality rate census* or it can be completely a different label like *current studies*.

In this section, we present the labeling algorithm which does the following:

Given a list of cohesive words from the comments clusters, the algorithm finds the best possible label from either the topic words or from an external knowledge base

4.4.1 Algorithm

We explored two different algorithms for labeling.

Topic Coherence Labelling: A set of statements or facts is said to be coherent, if they support each other [32]. Coherence measures can detect how much each fact correlates with each other based on an overall coherence score. We exploit this fact to select topic word as the label for the cluster. Algorithm 2 explains the general steps followed for selecting the topic word as the label.

Algorithm 2 Topic Coherence Labelling Algorithm

```
Require: topic word list, coherence metric
1: calculate topic coherence score for the list minus the specific word
2: Repeat steps 1 for each word in topic word list
3: Select the label with least coherence score when removed
```

To find the topic coherence we used an online too.[14] Different coherence metrics could be used to find the overall coherence of the topic words. Algorithm 2 assigns that topic word as label for which the gain in coherence is maximum when added to the list of existing topic words. These coherence scores are calculated based on word the co-occurrences in the English Wikipedia or any large corpus. These readings have been found to correlate with human ratings. To know more about the coherence measures refer [32].

Graph based Labelling: The problem with above approach is that it cannot create an abstractive label. For instance, the above approach works if you want to get *gun* as label for *gun, laws, shooting firing*, whereas in case if you want to get *color* for *red, blue, green, yellow* , it is not possible. To address this issue, we extended the topic labeling algorithm described in [33]. Through this approach some form of semantic abstraction is enabled for the labeling the topics. The algorithm 3 depicts the steps followed in the labeling process

Algorithm 3 Graph based Labelling Module

```
Require: topic word list, centrality measure, graph expansion relations
1: Create topic-wise graph based on relations defined by dbpedia
2: Expand the graph to 2 hops using these relations
3: Use centrality measures to find the central node of the graph topology
```

We found the following dbpedia relations to be useful in expanding the graph: *SKOS.broader*, *SKOS.broaderOf, RDFS.subClassOf, DCTERMS.subject*. This had better coverage with lesser spurious expansion compared to what is being mentioned in [33]. The expansion of concepts creates a large graph. As per [33], the central node of the graph thus created should represent a cohesive, yet abstractive label for all the concepts that it encompasses. To extract the label, graph centrality measures are used. The following centrality measures were used in finding the central node which would form the label for the topic mentioned: *betweeness_centrality*, *load_centrality*, *degree_centrality*, *closeness_centrality*.

4.4.2 Evaluation

We evaluated this system using the dataset provided by [34]. This dataset contains 228 topics annotated with labels by 6 annotators. Each topic contains 10 topic words. We selected the best label from the annotation set for each topic using a majority voting scheme and then manually compared the topic label generated by our module. Two evaluations were performed:

[14] http://aksw.org/Projects/Palmetto.html

1. Best system generated topic label was compared with the topic label provided by the annotation set.
2. Top 5 system generated topic labels were compared with the topic label provided by the annotation set.

Based on the evaluation we categorized them in to 'inappropriate', 'related', 'somewhat good', 'very good' as mentioned in [34]. The evaluation of Graph based Labelling algorithm is performed in this manner. In case of Topic Coherence Labelling algorithm, this evaluation is cannot be performed as we found that all topic words can be categorized as *related* by the annotator.

The result of our evalation is presented in the Table 9 and the Table 10.

Table 9: Evaluation by analyzing the best system generated label with the best label given by annotator

Label Annotation	Number of labels	Percentage Match
Inappropriate	44	19.20%
Related	84	36.84%
Somewhat Good	64	28.07%
Very Good	36	15.78%

Table 10: Evaluation by analyzing the top 5 system generate labels with the best label given by annotator

Label Annotation	Number of labels	Percentage Match
Inappropriate	21	9.21%
Related	27	11.80%
Somewhat Good	113	49.50%
Very Good	60	26.31%

We observed that labelling is coherent and if we select top 5 labels then coherency improves. The abstraction is able to capture the semantics of the topic words. However it could be improved (more focused, specific and less generic) by selecting the best relation in dbpedia to expand. We would be focusing our efforts towards this.

4.5 Extractive Summarization

Due to the availability of social media sites and the exponential growth of Internet use, online users communicate and share their opinions in textual form in online media. News sites are one example of the media in which users express their opinions in form of comments about the topics published in online news articles. As more and more content is published, the amount of such comment data increases. It therefore becomes difficult for readers and potential discussion participants to easily or quickly digest and understand the overall details in controversial discussions. Automatic text summarization is a promising technique to address this problem by helping users to digest the information on the web.

In this section we outline the initial algorithms we have developed to summarize news article comments and present our evaluation method and results.

4.5.1 Algorithms

We have explored a range of algorithms for comment summarization that may be loosely grouped into three classes: baseline algorithms that do not make use of any language processing techniques, basic text processing-based techniques that make use of limited language processing techniques (text similarity measures only) and clustering and linking approaches that make use of the comment-article linking and clustering techniques described in previous sections above.

Baseline summary generation We have implemented several baseline methods. These methods do not require language processing techniques but are generated based on metadata available in the news-comment data. The following are our baseline methods:

- **Time-First-Comment-in-Thread:** As mentioned in Section 4.3, news comments are mostly organized by threads. Ideally a thread contains comments on the same topic. Each comment within the thread has associated meta-data such as the date of publication. In this baseline method we sort threads based on the publication date of the first comment in them. Once such sorting is performed we visit the threads from top to bottom and select the first comment from each thread to include in the summary. This is performed until the summary length threshold is violated, e.g. until the summary contains 250 words.
- **ParticipantCount-First-Comment-in-Thread:** This is again similar to the first baseline. The only difference is that the sorting is performed based on the number of unique participants (i.e. commenters) within the thread. After sorting the first comments from the threads are included in the summary until the summary length threshold is reached.
- **CommentCount-First-Comment-in-Thread:** In this baseline we sort the threads based on the number of comments they contain. After sorting, the first comment from threads (top-to-bottom) is included in the summary.

Basic text processing-based summary generation Here we describe our first implementation of a summarizer that integrates some text processing to generate a reader comment summary. We distinguish between systems that are purely based on text similarity and those that do make use of linking and clustering components. The purely text-based approaches are listed below:

- **Article-Sim-First-Comment-in-Thread:** We take the centroid of each thread and compute its similarity to the article using cosine similarity. The value of the cosine measure is used to sort the threads from most to least similar. After sorting the threads, the first comment from each thread, starting from the first thread, is included in the summary.

- **Article-Sim-CentroidClosest-Comment-in-Thread:** Similar to the previous, but this time comments closest to the thread centroid, rather than the first comment in the thread, are included in the summary.
- **ArticleLead-Sim-First-Comment-in-Thread:** Similar to Article-Sim-First-Comment-in-Thread but instead of comparing with the entire article we use the lead part of the article (e.g. the first 5 sentences).
- **ArticleLead-Sim-CentroidClosest -Comment-in-Thread:** Similar to Article-Sim-CentroidClosest-Comment-in-Thread but instead of comparing with the entire article we use the lead part of the article (e.g. the first 5 sentences).
- **Time-CentroidClosest-Comment-in-Thread:** As in the Time-First-Comment-in-Thread baseline we again sort the threads based on the publication date of their first comment. But then we select the comment that is most similar to the centroid of the thread. We use cosine similarity to compute the similarity between a comment and the centroid of the thread.
- **ParticipantCount-CentroidClosest-Comment-in-Thread:** This is similar to the ParticipantCount-First-Comment-in-Thread baseline but we use centroid similarity to determine the comments to be included in the summary.
- **CommentCount-Centroid-Comment-in-Thread:** Similar to the CommentCount-First-Comment-in-Thread baseline but this time comments closest to the centroid are considered the summary-worthy ones.

Clustering and Article-Linking Approaches These are approaches that make use of comment-article linking and comment clustering. They first require clusters of comments which are obtained using the clustering approach described in Section 4.3.1. Once the clusters are generated we determine which of the clusters link back to the article. For this we make use of the linking approach described in Section 4.2.3. Note that we consider an entire cluster to be linked to the article (article segment) when any of its comments are linked to the article. With this method we obtain clusters that are linked to the article and clusters which are not. The third group of clusters is the union of the other two, i.e. one that contains all clusters. Based on these three groups of clusters we generate four different summaries:

- **Linked-Cluster-ArticleSim-Summary:** Uses only clusters linked to the article. We sort the clusters using their similarity to the entire article. To compute the similarity we take the vector representation of the centroid of the cluster and compute its cosine angle to the vector representation of the article. Once sorting is finished, we start from the most similar cluster and take from each cluster the comment that is closest to the cluster centroid. Again we apply summary length threshold when including the comments in the summary.
- **Linked-Cluster-ArticleLeadSim-Summary:** Similar to the previous, but instead of using the entire article only the lead part (first 5 sentences) of the article are used to sort the clusters.
- **Un-Linked-Cluster-ArticleSim-Summary:** Same as the first, but works only using the clusters that do not have links to the article segments.

Table 11: Summary evaluation results.

System	R1	R2	R-SU4
Human-Human	0.41	0.07	0.13
Time-First-Comment-in-Thread	0.35	0.04	0.10
Time-CentroidClosest-Comment-in-Thread	0.35	0.04	0.10
ParticipantCount-First-Comment-in-Thread	0.36	0.04	0.10
ParticipantCount-CentroidClosest- Comment-in-Thread	0.37	0.04	0.11
CommentCount-First-Comment-in-Thread	0.37	0.04	0.11
CommentCount-Centroid-Comment-in-Thread	0.37	0.04	0.11
Article-Sim-First-Comment-in-Thread	0.39	0.04	0.12
Article-Sim-CentroidClosest- Comment-in-Thread	0.39	0.04	0.12
ArticleLead-Sim-First-Comment-in-Thread	0.41	0.05	0.12
ArticleLead-Sim-CentroidClosest- Comment-in-Thread	0.42	0.05	0.13
Linked-Cluster-ArticleSim-Summary	0.37	0.04	0.11
Linked-Cluster-ArticleLeadSim-Summary	0.40	0.04	0.12
Un-Linked-Cluster-ArticleSim-Summary	0.24	0.02	0.06
All-Cluster-ArticleSim-Summary	0.36	0.04	0.10

- **All-Cluster-ArticleSim-Summary:** Same as above but works on all clusters – clusters containing linked and unlinked ones.

4.5.2 Evaluation

As with cluster evaluation, to assess the quality of extractive summarization we use the gold standard summaries created by USFD. This data is described in Section 4.1.7 above.

ROUGE In the assessment we compared the automatically generated summaries against model summaries written by humans using ROUGE [35]. Following the Document Understanding Conference (DUC) evaluation standards we used ROUGE 1 (R1), ROUGE 2 (R2) and ROUGE SU4 (RSU4) as evaluation metrics [36] . ROUGE 1 and 2 give recall scores for uni-gram and bi-gram overlap respectively between the automatically generated summaries and the reference ones. ROUGE SU4 allows bi-grams to be composed of non-contiguous words, with a maximum of four words between the bi-grams.

Results The results of the summary evaluation are shown in Table 11. From the table we can see that the best system is the **ArticleLead-Sim-CentroidClosest-Comment-in-Thread** system. In case of R1 it agrees with 42% with both human summaries. This is higher than the R1 agreement between the humans (**Human-Human**). The R2 and RSU4 scores are identical. The next best results are obtained with the **Linked-Cluster-ArticleLeadSim-Summary** system. Note this system makes use of linking and clustering. Its other variant – textbfLinked-Cluster-ArticleSim-Summary, which makes use of entire article to sort clusters – scores worse than the variant that makes use only the lead part of the article (first 5 sentences). The worst results are obtained when the summaries are generated based on un-linked clusters (**Un-Linked-Cluster-ArticleSim-Summary**).

Table 12: Example summaries.

Human summary	ArticleLead-Sim-CentroidClosest-Comment-in-Thread summary
The article evoked a lot of passion from commenters and there were a number of debates/divides. The most prevalent was the disagreement over ”the poor choosing to be poor” and how people should blame their own choices. A lot of commenters were fiercely against this. The debate was less centred around the difference between the 70s and 80s in terms of housing, but more about those who struggled versus those who haven’t struggled, regardless of the decade. There was a big discussion in which commenters who bought homes in the 70s/80s said they too have struggled, just like people now. Many of them expressed discontent over pensions and the retirement age being unfairly increased. Others said they have paid off their mortgage, but will never make money on it or that taking care of adult children has hit them financially. There was some anger toward those who are ”sitting comfortably” property-wise, yet complaining about other problems. All commenters seemed to agree that houses are harder to buy now. Some provided different reasons for this: high rent/interest rates and decreasing incomes. People were concerned for University students and the debts they have, stating the taxpayer should cover them. Some commenters tried to remind people that everybody is struggling, no matter what demographic they fall under. A few commenters were unimpressed with the article, stating it was not journalistic enough. They felt it was a ’regurgitation’ of statistics and seemed not to be fans of objective reporting of this matter.	Started work in late 70s, have secured appropriate qualifications -even invested in some more this year. I think of retiring and know I never will. Got mortgage in 2006, (post grad and appropriate qualifications for two sectors) and provide home for self and two students. I have applied for hundreds of jobs with necessary experience and qualifications to match, or more usually exceed stated requirements. Thanks to linkedin, I get to see how they prefer younger people for the roles. I have struggled for jobs in a number of recessions as primarily worked in the privater sector but this the worst. I am earning less now than I did in 1990 and will be hoping for a step up by queuing at one of Aldi’s infamous open day’s for new staff this Saturday. ” Therefore they ’choose’ not to be richer? I suspect for most of them choice doesn’t enter into it. Used to have to get up at midnight, half an hour before we went to bed... etc Because, believe me, given the choice I’d be working more and earning more. The fact my mortgage is paid off has allowed me to continue in my home, though it’s a struggle and we may have to sell. Bit of a sweeping generalisation. Many of us still have the costs of adult kids living at home, or are using retirement savings to help them find a place of their own. Not all older workers own, those that rent have seen rents increase while wages have at best stagnated. Of those that do own, many have seen their wages eaten up by inflation and low/no wage increases. None of this is new to anyone, all of us, young and old are in the same boat, stagnant wages and increasing living costs. Stop polarising.

Table 12 shows an example summary generated using the **ArticleLead-Sim-CentroidClosestComment-in-Thread** system. The table also shows an human summary.

Discussion Just how appropriate ROUGE is as a measure for assessing the reference summaries of reader comments written to match our summary definition (Section 2.2) is questionable (the low human-human ROUGE score may support this sceptical view). Our summary definition specifies abstractive summaries. Particular characteristics of these summaries include aggregation over comments (how many people held one position versus another) and the characterisation of perhaps subtly different perspectives on issues, where issues are themselves frequently identified by abstracting over several comments (i.e. they may not be explicitly stated in the text). ROUGE, which is fundamentally a lexical overlap measure, may not be well suited a to assess these characteristics of our summaries and hence may not be appropriate as an intrinsic evaluation measure for this type of summary. More investigation of this issue is needed. Nontheless, carrying out a ROUGE-based assessment is an informative first step in assessing our automtically generated summaries.

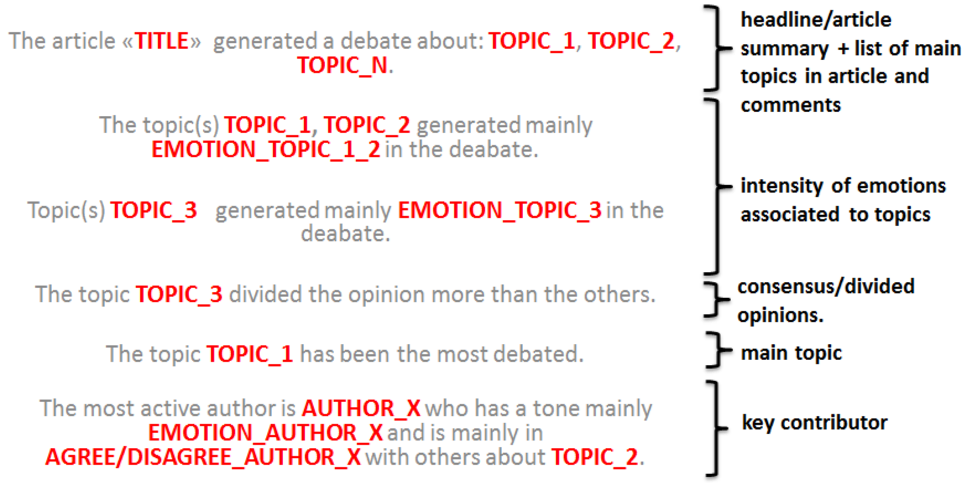
4.6 Template-based Summarization

A strong selection of the information available in social media conversations, driven by a theorethical model, allows us also to direct information into a template-based summary. We implemented a template-based summarization in Italian, driven by the thoretical model provided by the Town Hall Meeting use case scenario.

The Town Hall Meeting use case scenario requires the following information to be present in the summary:
1) headline/article summary,
2) key contributors,
3) list of main topics in the article and comments,
4) intensity of emotions associated to topics,

5) consensus or divided opinion on topics.

According to these, we designed a template-based summary that includes the parts depicted in Figure 10:



The article «TITLE» generated a debate about: TOPIC_1, TOPIC_2, TOPIC_N.

The topic(s) TOPIC_1, TOPIC_2 generated mainly EMOTION_TOPIC_1_2 in the deabate.

Topic(s) TOPIC_3 generated mainly EMOTION_TOPIC_3 in the deabate.

The topic TOPIC_3 divided the opinion more than the others.

The topic TOPIC_1 has been the most debated.

The most active author is AUTHOR_X who has a tone mainly EMOTION_AUTHOR_X and is mainly in AGREE/DISAGREE_AUTHOR_X with others about TOPIC_2.

headline/article summary + list of main topics in article and comments
intensity of emotions associated to topics
consensus/divided opinions.
main topic
key contributor

Figure 10: Template-based summary schema. In red and bold there are the valiable fields to be filled in, in black the fixed template.

The template includes fixed parts (in black) and variables computed by the algorithm (in red). These parts are extracted and processed by different algorithms, described in the next section. The Template-based summarization has been tested only intrisically, evaluating the single modules, We did not evaluate the summary extrinsically.

4.6.1 Algorithms

From a technical point of view, the template is filled with data from three different modules: topic extraction, mood prediction and agreement/disagreement detection. In addition to this the template-based summary includes a small part of metadata from the article (article title). The pipeline of the system, depictd in Figure 11, consists of different modules: topic extraction, agreement/disagreement detection, mood detection and comment linking and template filling.

- **Topic extraction**: the module for topic extraction is unsupervised, based on Hierarchical Latent Dirichlet Allocation [37] [38]. This module takes as input article and comments and outputs a topic model. Topics are extracted from the first level of the hirarchy. We evaluated the quality of the topics at sentence level exploiting the annotation notes from the annotation of Argument Structures. The results showed that “wrong topic” occurs 13.5% of the times.
- **Agreement/Disagreement detection**: the module for the prediction of Agreement/ Disagreement

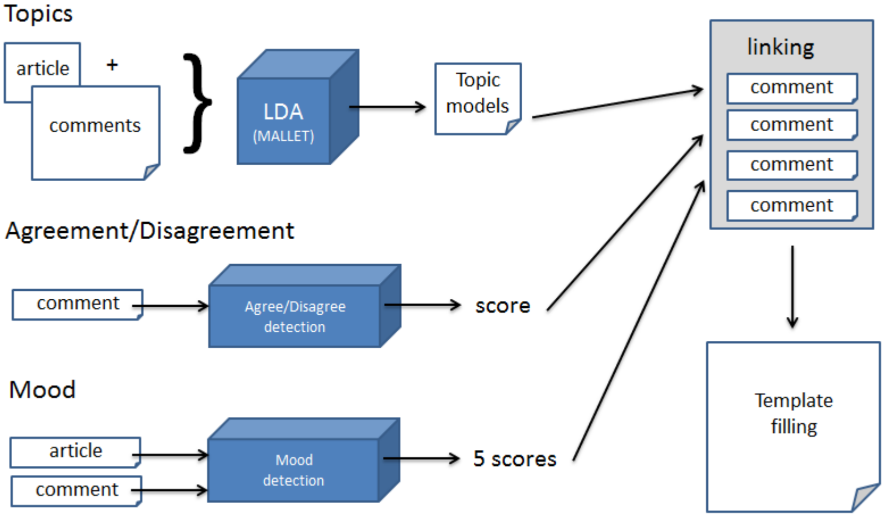


Figure 11: Template-based summary schema. In red and bold there are the valiable fields to be filled in, in black the fixed template.

is supervised, based on a cross-language model trained on Italian (CorEA Corpus), because the agreement/disagreement relation have been annotated manually as part of the Argument Structure described in Deliverable 4.2. The module for Agreement/Disagreement extracts from raw text a vector of 84 shallow statistical dimensions about text encoding, characters, ngrams, punctuation, numbers, parentheses, uppercases, lowercases, word freq, word length, string similarity, emoticons, parentheses, tf*idf, similarity of uppercase words and sine of the frequency of word pairs. We normalized all the features and trained the system using a 66% of the data and tested it on 33%, We performed feature selection searching for the subset of features with the highest individual predictive ability and the lowest degree of redundancy [39]. We trained a Support Vector Regressor [40], obtaining a Mean Absolute Error (MAE) of 0.42 over a majority baseline (score mean) of 0.44.

- **Mood detection**: the module for mood detection has been trained on Italian, on the CorEA corpus. We selected this corpus because it contains metadata about mood of the readers. The algorithm exploits all the 84 shallow statistical dimensions described above for Agreement/Disagreement detection: text encoding, characters, ngrams, punctuation, numbers, parentheses, uppercases, lowercases, word freq, word length, string similarity, emoticons, parentheses, tf*idf, similarity of uppercase words and sine of the frequency of word pairs. The learning algorithm is a Support Vector Regression and the settings for the evaluation are 10-folds cross validations and Root Mean Squared Error as evaluation metric. The moods defined in CorEA are: “amused, “disappointed”, “worried”, “indignated”, “satisfied”. Table 13 reports the results of the intrinsic evaluation of Moods. We also compute the overall mood of each single blogger by aggregating the mood predictions on all comments wrote by the blogger.

- **Linking**: this module matches predicted agreement/disagreement, predicted mood and topics to each comment and authors of the comments (contributors). This is metadata and no evaluation

Table 13: Evaluation of mood detection.

Mood	Art. baseline(RMSE)	Articles (RMSE)	Comm. baseline (RMSE)	Comments (RMSE)
Amused	0.8	0.12	0.13	0.14
Disappointed	0.13	0.17	0.13	0.15
Indignated	0.36	0.27	0.35	0.24
Satisfied	0.31	0.27	0.24	0.20
Worried	0.10	0.14	0.10	0.15
avg	0.20	**0.19**	0.19	**0.18**

is required.

5 Proposed Final Conversation Analysis / Summarization Outputs

In this section we describe the proposed additions/refinements to the existing outputs for the use cases already being addressed.

5.1 Refinements to Speech Conversation Analysis / Summarization Outputs

For the speech usecase, we have described two approaches. The first one uses hand-written templates for generating synopses and focuses on generating training data for slot recognition. The second one reimplements [4] and adapts it to constraints of the SENSEI data.

While both approaches have been applied on one language, we plan on extending them to the other languages of SENSEI, and particularly to assess and limit the resources required for such porting. We will also merge the approach to benefit both from template generation and robust slot filling and in particular, we will work on improving the coverage of the generated templates through the use of annotations generated by other WPs such as semantic parses and argument structure.

We have started working on deep learning approaches for template generation which use recurrent neural networks for learning how to generate templates from a representation of the conversation. While these approaches require large amount of training data, we will explore ways of adapting them to low-resources domains such as call-centre conversations.

In the context of the extrinsic evaluation, we plan on creating synopsis templates for generating synopses that expose para-semantic features of the conversations, such as ACOF questions an emotions, in order to assess in fluent text the status of the agent for use by call-centre supervisors.

Finally, we plan on converging the work on the speech use case with that of the social media use case, by drawing from similarities between the two media and building cross-media systems or cross-fertilizing the work between use cases.

5.2 Refinements to Social Media Conversation Analysis / Summarization Outputs

With respect to social media analysis and summarization we plan to work in two directions. First we will extend and improve what has been described above as the extractive approach to producing summaries as defined in Section 2.2. Second we will explore one of the other use cases proposed in D1.2 and highly rated by users, specifically Use Case 5 “Identifying Trends in Reader Comments”.

To extend and improve the extractive approach to “Town Hall Summary” like summaries we intend to do several things:

- We will seek to incorporate discourse information, as supplied by components developed in WP4, specifically co-reference information and discourse relations, as provided by the discourse parser. Co-reference information should help to improve clustering since, for example, pronominal references, which provide little information on their own, can be replaced by their antecedents, supplying substantially more information to the clustering algorithm. Discourse relations should help us to determine relations of agreement and disagreement between comments, which is necessary if our summaries are to move closer to the target summary type than our summary definition specifies.
- We will seek to incorporate parasemantic/affective information, as supplied by components developed in WP3, to allow us to include this informatiom in summaries and representations of comment clusters (e.g. to graphically signal clusters containing more heated exchanges). This information is not currently in the summaries produced by the extractive approach.
- We will seek to move away from simplistic extractive summaries that extract top ranked comments from clusters towards summaries that more closely match our target summary type by (a) attempting to identify and include in summaries alternative perspectives on particular issues (b) attempting to quantify, informally, the number of commenters commeting on particular issues or taking particular sides on particular issues. As part of this work we will investigate alternatives to ROUGE as a measure of intrinsic quality of our summaries, measures that will be better able to assess the distinctive aspects of the summaries we aim to produce (i.e. summaries that address points (a) and (b)).

Of course we will continue to incrementally refine the current approach as well, taking into account feedback from both the extrinsic and intrinsic evaluations just recently completed.

To explore the “Identifying Trends” use case we need to address several technical challenges:

- We need to develop techniques for clustering comments across comments sets for multiple articles. These may be straightforward extensions to approaches we have already used for single comments sets, such as clustering over merged comments sets from multiple articles; or they may involve new approaches, such as building clusters of clusters.

- We need to explore how best to model and present to users the temporal/dynamic aspect of cluster evolution.

Ideally we will have some “Identifying Trends” technology available for inclusion in the final SENSEI prototype. We not expect to be able to carry out an extrinsic evaluation of any technology developed for this use case, due to (a) uncertainties about what will be developed and how it might be evaluated and (b) lack of resources to evaluate both the final“Town Hall Meeting” technology and the new “Identifying Trends” technology. However we will try to solicit feedback from users on any technology we do develop.

6 Conclusions

In this deliverable we have first outlined what SENSEI understands speech and social media summaries to be. For both speech and social media use cases we have presented initial conversation analysis and summarization components and reported intrinsic evaluation results for them. Since some of the components are still in relatively early stages of their development, they require more effort to realise SENSEI’s vision for summarization and analysis of conversational data, in particular incorporating the outputs of other workpackages, such as WP3 and WP4 that are now reaching maturity. Along thess lines we have also outlined work to be done for Period 3 in relation with the summarization components’ extensions and refinements.

References

[1] Mohamed Morchid, Mohamed Bouallegue, Richard Dufour, Georges Linares, Driss Matrouf, and Renato De Mori. An i-vector based approach to compact multi-granularity topic spaces representation of textual documents. In *the Conference of Empirical Methods on Natural Lnguage Processing (EMNLP)*, 2014.

[2] Kalina Bontcheva, Leon Derczynski, Adam Funk, Mark A Greenwood, Diana Maynard, and Niraj Aswani. Twitie: An open-source information extraction pipeline for microblog text. In *RANLP*, pages 83–90, 2013.

[3] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537, 2011.

[4] Tatsuro Oya, Yashar Mehdad, Giuseppe Carenini, and Raymond Ng. A template-based abstractive meeting summarization: Leveraging summary and source text relationships. In *Proceedings of International Conference on Natural Language Generation (INLG 2014)*, 2014.

[5] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[6] Ashley A. Anderson, Dominique Brossard, Dietram A. Scheufele, Michael A. Xenos, and Peter Ladwig. The nasty effect: online incivility and risk perceptions of emerging technologies. *Journal of Computer-Mediated Communication*, 2013.

[7] Nicholas Diakopoulos and Mor Naaman. Towards quality discourse in online news comments. In *Proceedings of the ACM 2011 Conference on Computer Supported Cooperative Work*, CSCW '11, pages 133–142, New York, NY, USA, 2011. ACM.

[8] Arthur D. Santana. Virtuous or vitriolic. *Journalism Practice*, 8(1):18–33, 2014.

[9] Meishan Hu, Aixin Sun, and Ee-Peng Lim. Comments-oriented document summarization: understanding documents with readers' feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 291–298. ACM, 2008.

[10] Elham Khabiri, James Caverlee, and Chiao-Fang Hsu. Summarizing user-contributed comments. In *ICWSM*, 2011.

[11] Yue Lu, ChengXiang Zhai, and Neel Sundaresan. Rated aspect summarization of short comments. In *Proceedings of the 18th international conference on World wide web*, pages 131–140. ACM, 2009.

[12] Zongyang Ma, Aixin Sun, Quan Yuan, and Gao Cong. Topic-driven reader comments summarization. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 265–274. ACM, 2012.

[13] Clare Llewellyn, Claire Grover, and Jon Oberlander. Summarizing newspaper comments. In *Eighth International AAAI Conference on Weblogs and Social Media*, 2014.

[14] Dyut Kumar Sil, Srinivasan H Sengamedu, and Chiranjib Bhattacharyya. Supervised matching of comments with news article segments. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 2125–2128. ACM, 2011.

[15] Mrinal Kanti Das, Trapit Bansal, and Chiranjib Bhattacharyya. Going beyond corr-lda for detecting specific comments on news & blogs. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 483–492. ACM, 2014.

[16] Mārcis Pinnis, Nikola Ljubešić, Dan Ştefănescu, Inguna Skadiņa, Marko Tadić, and Tatiana Gornostay. Term extraction, tagging, and mapping tools for under-resourced languages. In *Proceedings of the 10th Conference on Terminology and Knowledge Engineering (TKE 2012), June*, pages 20–21, 2012.

[17] G. Salton and M. Lesk, E. Computer evaluation of indexing and text processing. In *Journal of the ACM*, volume 15, pages 8–36, New York, NY, USA, 1968. ACM Press.

[18] Peter Kolb. Experiments on the difference between semantic similarity and relatedness. In *Proceedings of the 17th Nordic Conference on Computational Linguistics-NODALIDA09*, 2009.

[19] Dekang Lin. Automatic retrieval and clustering of similar words. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 2*, pages 768–774. Association for Computational Linguistics, 1998.

[20] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[21] Hideya Kawaji, Yoichi Takenaka, and Hideo Matsuda. Graph-based clustering for finding distant relationships in a large set of protein sequences. *Bioinformatics*, 20(2):243–252, 2004.

[22] Zhiwen Yu, Hau-San Wong, and Hongqiang Wang. Graph-based consensus clustering for class discovery from gene expression data. *Bioinformatics*, 23(21):2888–2896, 2007.

[23] Günes Erkan and Dragomir R Radev. Lexrank: graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, pages 457–479, 2004.

[24] Adam Schenker, Mark Last, Horst Bunke, and Abraham Kandel. Comparison of distance measures for graph-based clustering of documents. In *Graph Based Representations in Pattern Recognition*, pages 202–213. Springer, 2003.

[25] Satu Elisa Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007.

[26] Clare Llewellyn, Claire Grover, and Jon Oberlander. Summarizing newspaper comments. In *Eighth International AAAI Conference on Weblogs and Social Media*, 2014.

[27] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[28] Stijn Marinus Van Dongen. Graph clustering by flow simulation. 2001.

[29] 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.

[30] 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.

[31] 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.

[32] Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the space of topic coherence measures. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pages 399–408. ACM, 2015.

[33] Ioana Hulpus, Conor Hayes, Marcel Karnstedt, and Derek Greene. Unsupervised graph-based topic labelling using dbpedia. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, pages 465–474, New York, NY, USA, 2013. ACM.

[34] Jey Han Lau, Karl Grieser, David Newman, and Timothy Baldwin. Automatic labelling of topic models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 1536–1545, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[35] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, volume 8, 2004.

[36] Hoa Trang Dang. Overview of duc 2005. In *Proceedings of the document understanding conference*, 2005.

[37] Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. Hierarchical dirichlet processes. *Journal of the american statistical association*, 101(476), 2006.

[38] Andrew K McCallum. Mallet: A machine learning for language toolkit. Technical report, 2002.

[39] Mark A Hall and Lloyd A Smith. *Practical feature subset selection for machine learning*. Springer, 1998.

[40] Shirish Krishnaj Shevade, S Sathiya Keerthi, Chiranjib Bhattacharyya, and Karaturi Radha Krishna Murthy. Improvements to the smo algorithm for svm regression. *Neural Networks, IEEE Transactions on*, 11(5):1188–1193, 2000.

Appendix A Guidelines for Writing THM Summaries

A note on the guidelines

We note that the following report on the guidelines for writing Town Hall Meeting Summaries is a working document, and later versions may follow (e.g. new examples may be added).

The guidelines are detailed and are primarily for reference. To help communicate the task, we run a training session, where we present and discuss the different stages of the task (as laid out in the guidelines).

In the training session, we also give annotators some exercises on using the interface for the writing task. This practice helps to ensure people are confident with all parts of the task, before allowing them to proceed to the proper annotation.

Confidentiality - this research is yet unpublished and we would appreciate it if people did not pass on the guidelines, thank you.

SENSEI research project team, Department of Computer Science, Department of Journalism Studies, University of Sheffield.

Contents

Guidelines for Writing SENSEI Summaries of Reader Comments (v1.4). USFD Mar. 2015

A.1 Introduction

Writing summaries of reader comments to news articles in social media is a difficult task. The sheer volume of comment, the complexity of debate and the format of online comment (e.g. where individual comments are organized into *threads*) presents a challenge for anyone attempting to get an overview of an entire community discussion.

To help people write summaries of reader comment we have developed a simple method, in 3 stages: ***comment labeling***, ***label grouping*** **and** ***summary generation***. (In a fourth stage, "*back-linking*", we ask people to link summary sentences to the supporting groups this information is useful to us to analyze the writing process, but is not part of the summary writing process).

Each stage is carried out in a specialised interface, which provides tools to support the summary writer (we sometimes refer to summary writers as "annotators" as we can view the summaries as a form of annotation on the set of comments being summarized). Instructions for labeling, grouping, generating and "back-linking" summaries are discussed in detail below. We first describe what the summary should look like, since it is this format that guides each of the individual stages.

A.1.1 The SENSEI Summary Format - the town hall meeting report

Given a news article and a set of reader comments, our aim is for an annotator to produce a short, mainly text-based summary of the discussion in the comments, similar to the way a reporter might take notes and write a report on a town hall meeting. An example town hall meeting (THM) style summary of a set of comments is shown on the next page, in Figure 12.

> **Tip**: Please look carefully at this example. It was written by an annotator who used our method and the annotation interface. We ask annotators to try to produce summaries that are similar to this format.

In a Town Hall Meeting there is typically an opening statement on a particular issue, followed by questions and discussion from the floor. We can view the news article as the opening statement and the reader comments as the public discussion, but we note there is no chair to guide the meeting, as you would see in a town hall setting.

A reporter attending a town hall meeting would ask the following kinds of questions. You may find it helpful to keep these in mind when producing your summary.

- ***"what were people talking about?"***
  - what were the main issues addressed in the comments?

  - what were the different views or perspectives on issues in the debate?

- ***"what issues did people agree/disagree about?"***
  - on what issues was there consensus/ divided opinion?

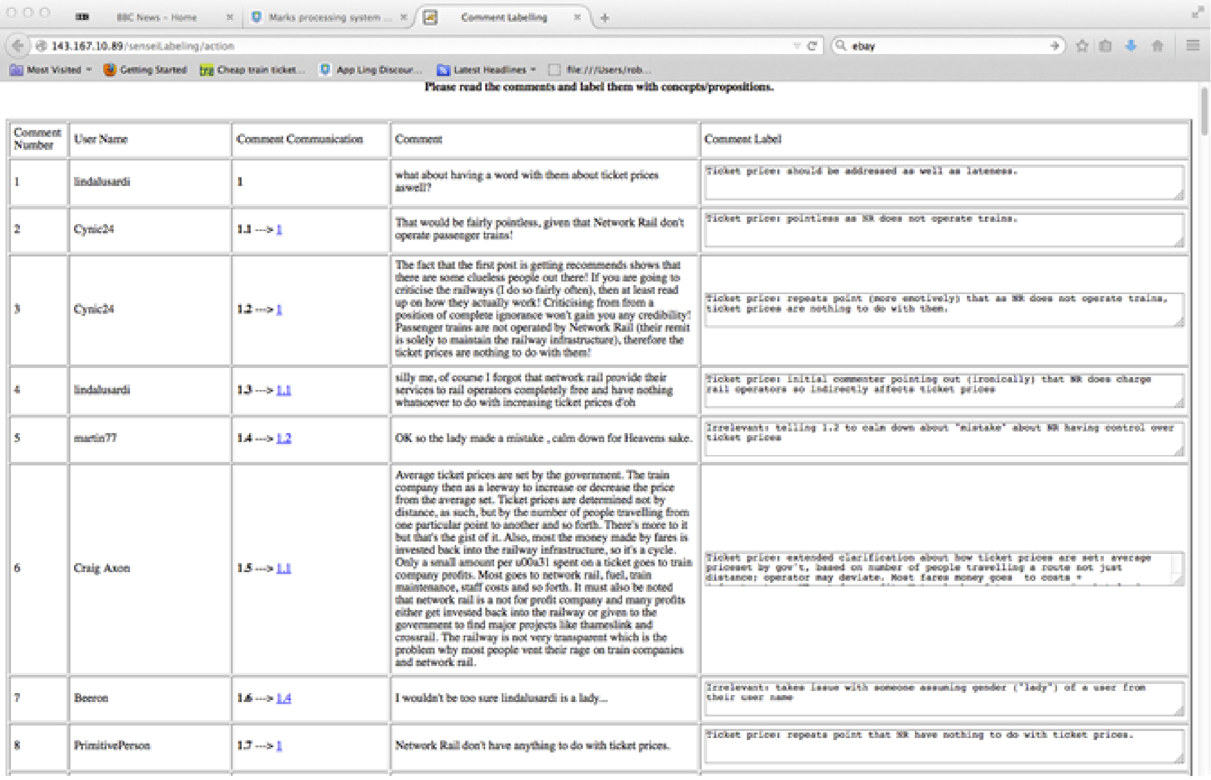
- ***"how did people feel about the issues? what issues did people feel strongly about?"***
  - what feelings were expressed on the issues? were there any intense feelings shown and on what issues?

- ***"how many/what proportion of commenters were talking about the issues? shared such views? shared such feeling?"***

If the summary were to closely resemble a town hall meeting report then it would include further information e.g.: a summary of the article, details/names of who took part, details of who was talking to who and who said what. However, in this task we are interested in a very simple type of summary, and while annotators should read the article and may find it helpful to note patterns of discussion between the commenters, we ask that details such as what the article was about and individual commenter names be left out of the summary.

> "The largest group of commenters discussed whether it makes sense to fine a public company like National Rail. Since NR relies on a public subsidy to run, fining it takes away the money it needs to deliver the service which either needs to be made up in increased subsidy or will result in lower service quality. Many suggested directors pay or bonuses should be cut instead; others suggested the fine could be part of a covert coalition plan to run the service down leading to performance issues which could be used as an argument for re-privatisation.
>
> Commenters also questioned how a fine would address congestion, the underlying cause of delays. Some pointed out that the train operating companies are also partly to blame for delays, but it was noted that there are separate penalties for them. Many questioned the point of using the fine to improve Wifi service on trains.
>
> One commenter proposed ticket price should be addressed as well as train lateness. This triggered many replies that NR does not set ticket price, rather the train operating companies do within a framework set by government. Some commented how the lack of transparency and division of roles between organizations involved in the rail service leads to confusion over where responsibilities lie. Calls for and against nationalization of the whole service formed another strand of the debate.
>
> Finally, a group of commenters complained about NRs planned extra investment in the south-east commuter network arguing that it already received disproportionate resource.
>
> End."

Figure 12: A Gold Standard SENSEI THM Summary Based on 100 comments to the article: http://www.theguardian.com/business/2014/jul/07/network-rail-fined-50m-pounds-late-trains (type: constrained length, 249 words, manual summary)



Please read the comments and label them with concepts/propositions.

Comment Number	User Name	Comment Communication	Comment	Comment Label
1	lindalusardi	1	what about having a word with them about ticket prices aswell?	Ticket price: should be addressed as well as lateness.
2	Cynic24	1.1 ---> 1	That would be fairly pointless, given that Network Rail don't operate passenger trains!	Ticket price: pointless as NR does not operate trains.
3	Cynic24	1.2 ---> 1	The fact that the first post is getting recommends shows that there are some clueless people out there! If you are going to criticise the railways (I do so fairly often), then at least read up on how they actually work! Criticising from from a position of complete ignorance won't gain you any credibility! Passenger trains are not operated by Network Rail (their remit is solely to maintain the railway infrastructure), therefore the ticket prices are nothing to do with them!	Ticket price: repeats point (more emotively) that as NR does not operate trains, ticket prices are nothing to do with them.
4	lindalusardi	1.3 ---> 1.1	silly me, of course I forgot that network rail provide their services to rail operators completely free and have nothing whatsoever to do with increasing ticket prices d'oh	Ticket price: Initial commenter pointing out (ironically) that NR does charge rail operators so indirectly affects ticket prices
5	martin77	1.4 ---> 1.2	OK so the lady made a mistake , calm down for Heavens sake.	Irrelevant: telling 1.2 to calm down about "mistake" about NR having control over ticket prices
6	Craig Axon	1.5 ---> 1.1	Average ticket prices are set by the government. The train company then as a leeway to increase or decrease the price from the average set. Ticket prices are determined not by distance, as such, but by the number of people travelling from one particular point to another and so forth. There's more to it but that's the gist of it. Also, most the money made by fares is invested back into the railway infrastructure, so it's a cycle. Only a small amount per u00a31 spent on a ticket goes to train company profits. Most goes to network rail, fuel, train maintenance, staff costs and so forth. It must also be noted that network rail is a not for profit company and many profits either get invested back into the railway or given to the government to find major projects like thameslink and crossrail. The railway is not very transparent which is the problem why most people vent their rage on train companies and network rail.	Ticket price: extended clarification about how ticket prices are set: average priceset by gov't, based on number of people travelling a route not just distance; operator may deviate. Most fares money goes to costs +
7	Beeron	1.6 ---> 1.4	I wouldn't be too sure lindalusardi is a lady...	Irrelevant: takes issue with someone assuming gender ("lady") of a user from their user name
8	PrimitivePerson	1.7 ---> 1	Network Rail don't have anything to do with ticket prices.	Ticket price: repeats point that NR have nothing to do with ticket prices.

Figure 13: Screenshot to show part of the comment labeling interface with completed label annotations. Comments sourced from: http://www.theguardian.com/business/2014/jul/07/network-rail-fined-50m-pounds-late-trains

A.2 Stage 1 - Comment Labeling

Having selected a news article and comment set to summarise, you may begin the task of *comment labeling*.

Definition: We view a label as a short text annotation, or note on an individual comment, which summarises the essential content in a comment.

For example, see a label for Comment 1:

> **Comment 1**: "*I can't wait, the plain packaging looks so cool, and i think it will encourage people to return to the days of stylish cigarette tins. Much better than advertising a brand when you smoke*".
>
> **Label**: "Plain packaging: cool; encourages return to stylish cigarette tins; preferable to advertising brands".

Labels provide the foundation for the summary writing task, and there are two main objectives for labeling:

1. To produce a note of key content in a comment so that you can group content and select content for use in a summary of the comment set.

   In the next stages (stage 2-grouping and stage 3-generating the summary) you will work mainly with your labels, and not the full comments. Your label should be sufficiently detailed to support further analysis, grouping and summary writing, without requiring you to refer back to the original comment (though you may do this if necessary).

2. Labeling should also help you to engage more deeply with the content of the comments, than you would, say, by simply reading the comments.

   By thinking carefully about what to put in a label and recording your thoughts, you should gain a good understanding of the comment. Labeling should also help people to remember details of a comment. These factors may help in future comment labeling, grouping and summary writing.

Note on reading the article: We advise annotators to ***read the news article before comment labeling***. This provides important background and will help you to make sense of what the comments are talking about. (The article is displayed in the interface.)

The Comment Labeling interface Annotators should label comments in the interface we provide for this task. A screen shot of part of the interface, showing comments and completed labels is shown in Figure 13.

Detailed instructions for labeling follow:

A.2.1 What to include in a label

- **Labeling should be guided by the goal of creating a SENSEI style summary of the comment set.** (See Section 1 above). The aim is to record content that you think might help you to create the summary or be included in the end summary.
  - In particular, try to identify important **topics, points, arguments or propositions** in the comment, and then make a note of the content in the label field.
  - If its obvious, please indicate if a person is for or against a particular issue.
  - Make a note of **jokes, strong feeling, and other emotional content** if you think it is a strong feature of a comment. (dont worry about doing this exhaustivelyrecord such aspects of a comment only if they stand out).
- **Labels do not need to indicate if the comment is referring to content in the article.** Read the article first so that you know what has prompted the discussion in the comments. But do not worry about recording whether a comment is referring to something said in the article.
- **Links between comments.** You might find it helpful to follow how a comment refers to another comment and to look for notable patterns of interchange.
  - E.g. is the same person making the same point in replies to many different commenters? Are lots of commenters taking issue with a commenter about a particular issue?

Make a note in your labels if you notice any patterns.

  - E.g. if a commenter is making the same point many times, you could note in the successive labels that he is "repeating the point".

- **Try and label ALL comments**, even the very brief comments, if you can make sense of what the comment is saying and if you believe it will be important to capture or count in the end summary.
  - E.g., **label brief comments** if they are simply agreeing with something said before.
- Longer comments may express multiple "points" or address different topics. In your label, include the content you think may be important to the end summary.
- You may find as you start labeling that it helps to **write out lots of details** in your label. But as you proceed through the comments you may find it easier to be more selective.
- In general, **ignore stuff** that you feel is irrelevant to the topic of conversation - e.g. remarks to other commenters about their political leanings, which do not make a comment on anything actually being said in the debate, e.g. "*well you Tories are all the same*". Also ignore comment which you cannot make sense of.

A.2.2 Forms of expression to use in a label

- Use **abbreviations and keywords** to indicate **key topics or themes**, but **also** ...
- Include **notes of any details** that you think may be important or may help you to remember what the comment says: they may come in useful when you group labels.
- Try **paraphrasing** the relevant part of the comment, rather than re-writing verbatim.
- You may always come back and review/update label descriptions later in the task.

Examples:

> Comment 1: "*I cant wait, the plain packaging looks so cool, and i think it will encourage people to return to the days of stylish cigarette tins. Much better than advertising a brand when you smoke.*"

Possible labels:

- "Plain packaging for cigarettes: likes; better not to have brands"
- "Plain packaging– in favour of"
- "PP-cool; glamourous"
- "Plain packaging: cool; encourages return to stylish cigarette tins; preferable to advertising brands".

A.2.3 Using similar terms to associate comments

As an annotator proceeds through the comment set, creating labels they go, he/she may see a new comment which appears to be addressing a topic or making a point which is the same as, or very similar to one seen during prior labeling. A key part of the labeling process is to recognize when this occurs.

Annotators should try to use **common terms** and expressions to describe such content in the respective labels. By doing so they make a link between comments, via the labels. This will be very helpful in stage 2 of the writing task, i.e. “Grouping”.

> **Tip**: Use of repeating “labels” = grouping or associating content.

When you recognize similar content:

- Create a label and include the **same keywords/expression** as used before, or create a new expression and use this in **both the new label and the previous label**.
- While using common words to indicate a link, annotators should remember **to distinguish any differences** between the comments.
- To save time when labeling big topics, (i.e. topics addressed by many commenters), try developing a **simple abbreviation or keyword**.

Examples of associated labels: Below are some labels for different comments discussing the topic of plain packaging for cigarettes and whether this will help to reduce cigarette sales and deter people from taking up smoking. The common terms in the labels indicate similarities between labels and the underlying comments, but there are also different points of view.

- Label a) “*plain packaging – cant see how it will change things*”
- Label b) “*plain packaging for cigarettes – reduces cigarette sales*”
- Label c) “*plain packaging – packaging only distinguishes brands; doesnt influence sales*”
- Label d) “*plain packaging for cigarettes – reduces cigarette sales*”
- Label e) “*plain packaging in Australia – suggests the sales reduce*”
- Label f) “*plain packaging in Australia – suggests the sales go up*”

So, labels b), d) and e) suggest a similar point of view is expressed in the underlying comments: that plain packaging will reduce sales. Labels e) and f) indicate contrary sides to an argument, in a similar context (in Australia-sales have gone up/ sales have reduced). Moreover, we can see that labels a) and c) are very similar since they both reflect the viewpoint that plain packaging will not lead to much change. A further useful revision would therefore be to include a new keyword e.g. “wont lead to change” so the revised labels would be:

- Label a) “plain packaging - won’t lead to change”
- Label c) “plain packaging - won’t lead to change-packaging only distinguishes brands; doesnt influence sales”

  Tip: finding similar content across different comments is the Eureka moment which will help to strengthen your feeling about what should go in a label and indeed what the core content for the summary will be.

Anaphora in comments. You may find very short comments that make reference to content which you have identified in previous labels. If you are confident about what a comment is referring to, then please label accordingly.

E.g., in Comment 3, we see an anaphor (“that”) referring back to a proposition in Comment 2. So the same expression is used in both labels, but with the added note that Comment 3 is agreeing with something.

> **Comment 2**: “*Plain packaging wont help people to stop buying cigarettes*”
> **Label**: “Plain packaging wont stop people buying cigs”
>
> **Comment 3**: “*I agree with that*”
> **Label**: “Agrees: plain packaging wont stop people buying cigs”
>
> **Tip**: Labeling such “lesser” comments is important since it will help to reinforce your understanding of the support for an argument in the commentshow many are for or against something etc.

A.2.4 Reviewing and revising your labels

As you proceed through the comment set, annotators should go back to review and, if necessary, revise their previous labels.

> **Please remember to do this from time to time, (e.g. every 25 or so comments) even if you dont have in mind a specific comment to re-consider.**

Reviewing and revising labels is important for a number of reasons:

- As we have seen in section 2.3 on associating labels, it is important to go back and **re-formulate previous labels** in order to help identify similar content in the comments.
- As you proceed through the comments your ideas about what content will be important in a summary may change. Reviewing gives you the opportunity to update your initial interpretation.
- A new piece of information in later comments may help you to more fully understand a previous comment or to see it from a different perspective.

A.3 Stage 2 - Grouping Labels

When comment labeling is finished, annotators should save and submit their labels. They then proceed to stage 2, and the task of "grouping labels".

Please note, once they have left stage 1, annotators should not return to continue editing labels in the stage 1 interface. They may only *view* the original comments and their labels via the link we provide at the top of the screen, which appears as:

"**For reference you may:**
Click to view the original comments and your labels in a new tab"

If annotators discover during grouping that their labels are insufficiently detailed or incorrect, they may make changes to their labels in the text box for stage 2.

The results of the stage 1, the collected labels, will be **displayed in the stage 2 interface window**.

A.3.1 Starting to Group

In stage 2 annotators should begin to **sort and group together similar or related labels**.

We note that grouping is a continuation of the process of using common expressions to associate labels.

> **Tip**: Grouping should be guided by the goal of creating a SENSEI style summary of the comment set. (See section 1 above). The aim is to identify groups that you think might help you to create the summary.

> **Tip**: You may find it helpful to break the set of 100 or so labels into initial groups e.g. of ten or so, or by thread, by adding line spaces, etc. You can then begin grouping similar content together.

The Add Group Tag button: This will insert a tag at the position where you leave the cursor. It will also add a prompt for a "Group Label".

Once you have identified a group you should add a "Group Tag", at the top of the group, or where you want the group to be, using the button provided.

You may delete, cut and paste Group tags, and move them around in your grouping file, as you wish.

> **Tip**: keep it flexible. As you continue with your analysis, you may move labels around between groups, add labels to groups, split groups and remove groups. Remember to update the Group Tags and Group labels accordingly

> **Tip**: It may feel useful to keep certain groups together in the text editor - e.g. if they feel loosely related. At some point they may be brought together into a single common Group.

Figure 14 shows an example of labels organised into groups. The summary shown in Figure 12 was written using some of these labels and Groups (see paragraphs 1, 2 and 4 in the summary).

A.3.2 Labeling Groups

It is helpful to give a *Group label* to indicate the basis for the group - i.e. to describe the common content.

You may identify a common **topic, argument or proposition, or humour, or strong feeling**.

You might also find it helpful to make a note of the different range of feelings within a group. Or different view points within the Group. If these appear to be expressed in a good number of labels this may be good reason to form a new sub-group within the group (see Section 3.3 below).

A.3.3 Sub-Groups

You may find as you gather labels into Groups that there are clusters of related labels within that Group. If you think it might be helpful to distinguish such clusters, you may create a Sub-Group. Please collect related comments together within the original Group and then add a Sub-Group tag and label for the Sub-Group at the top of the smaller cluster of labels.

For example, see the Sub-group in Figure 14.

Important: Sub-groups implicitly end when a new Group or Sub-group begins. Therefore, if you insert a Sub-group in the middle of an existing Group then any comment labels following the Sub-group label and up until the next Group or Sub-group label will be assumed to be in the new Sub-group. In particular, blank lines between labels will not be interpreted as indicating the end of a Sub-group. So, when adding Sub-groups please check over the labels in the parent Group/Sub-Group and if necessary move labels around to ensure they are in the correct position.

A.3.4 Labels in multiple Groups

Often a label may mention multiple topics or issues and therefore may figure in different groups. For such cases annotators should make duplicate copies of the label and then include under different groups as required. (For example in Figure 14, the label for comment 79 is included in two groups, “Group : Not fair that SE gets the investment” and “Group : Why use the fine on wifi?”).

A.3.5 All labels should be put into at least one Group

- **Single labels** Single labels may be placed in a “Group;; or “Sub-Group” if you think the content is important and may contribute to the debate. Annotators should remember to add a Group/Sub-Group tag and label.

- **The Miscellaneous Group** Please include all labels that you have not placed in a Group into a “Miscellaneous Group” - we have already added a tag for convenience. This is where to put comments you have not been able to understand or irrelevant remarks that make no contribution to the debate.

A.3.6 Using Groups and Labels to write your summary

When selecting content to go in the final summary an annotator need not refer to all of the groups, sub-groups or all labels collected. I.e. content collected during the grouping stage may be left out of the summary.

A.4 Writing the summary

When annotators are satisfied with their grouping they should proceed to the third stage of the task: to produce a written summary of the comments. The summary should be based on their analysis in all previous stages, using the Groups, labels and possibly the original comments.

> **Tip**: Annotators should stand back and read over the results of stage 2 grouping (Groups and comment labels), before attempting to write.

Further analysis and content selection may take place as a person decides on the text that should go into the summary. When formulating a summary sentence an annotator may:

- Develop the original group label into a more polished sentence.
- Use content from within a groupsay a particular detail from a label or comment that they think helps to exemplify a point or make something clear in the summary.
- Carry out further grouping in their heade.g., the most effective way to summarise content from two separate groups might be to create a new high-level, inclusive statement based on the two groups.

A.4.1 Ordering content

We do not provide detailed instructions on how to order content, but the summary should be well formed and readable. Annotators should think about the order in which they present information and should do what makes a ‘good’ summary.

A.4.2 Counting Content

Getting rough estimates of the quantity of comments talking about the various topics and issues; and in terms of the extent of opinion and feeling shown toward different issues, is an important part of the process of developing the content to go into the summary.

Annotators should think about counting the labels in the groups that inform their summary and then use an appropriate quantifier, but we do not insist that people provide an exhaustive or precise count of labels and comments in the summary

(Figure 3 shows an exception where *during grouping* an annotator has counted the number of different commenters in a group. You may find this helpful, but going into such detail is optional. It is usually possible to assess counts of labels in your head based on the saved and displayed "grouping" file.)

To keep the process "light and easy" we recommend you use quite general expressions when referring to quantities. **Example phrases for quantifiers:**

- "*Most of* the comments"; "*The majority of* comments";
- "Opinion was *divided*";
- "*at least*";
- "*in the minority*";
- "*outliers*", "*exceptions*";
- "*a few*", "*several*";
- "*all*";
- "*many*", "*a substantial number of*", "*some*", etc.

A.4.3 Summary length

We instruct summary writers to create 2 different summaries:

1. ***Unconstrained summary***: a summary, of a comfortable length, written with the knowledge that the final aim is to produce a summary of around 150-250 words.

   > **Tip**: Once you have an initial draft of a summary, carry out **word-smithing** as best as possible. I.e. reduce the total number of words by finding more concise forms of expression to capture content, but without losing any detail or actual content.

   Note the word count. Save and submit this text as an "*unconstrained summary*".

2. ***Constrained summary***: a summary of restricted length, minimum 150 to maximum 250 words.

> **Tip**: read over your unconstrained summary and identify the most important and least important content. You may then re-arrange content, delete content, continue word-smithing, and revise sentences until you reach the target length.

When you are happy with your summary and it is within the required word limit, you may save and submit the text as the "*constrained summary*".

A.4.4 Returning to previous stages

The 3 different stages in the writing and annotation task are carried out more or less in succession. Once an annotator has submitted work in a stage and proceeded to the next stage, he must not go back and revise his previous annotations. However,

- In stages 2 and 3, annotators may view the final results of the comment labeling screen in a separate tab, in order to refresh their memory of the detail of a comment or their original label.
  - Note: Annotators may continue to edit their labels in stage 2.
- In stage 3, if an annotator discovers he has proceeded from the grouping stage by mistake he may leave stage 3 and return to stage 2 to continue grouping, by following the special instructions.

A.5 Stage 4: Back-Linking sentences to labels

We would like you to link the sentences from your final summary with the Group(s) of labels, and/or Sub-group(s), that helped you to write that sentence.

I.e. where possible, we would like you to link the summary sentence to the **Group(s) and/or Sub-group(s)** that support that sentence.

You must first select a sentence in the summary by clicking on it (it should be shown in a green colour highlight). Then you check-mark any relevant groups from the set shown below on the screen.
When you have made your selection you should save the links and proceed to select a new sentence.
If you change your mind you may go back and add/remove previous saved links, but remember to save the changes.

> **Tip**: If a sentence was based on information in multiple Groups link to all relevant Groups.

> **Tip**: Linking to a Group implies that at least some of the labels in that Group have played a part supporting the sentence, (i.e. not all labels in a Group need to have played a role for their to be a link).

LINKING to Sub-groups

Tip: If the labels that support a summary sentence are localized to one or more Sub-groups, then you may select only those specific Sub-groups. Otherwise, select the entire Group.

You should not select both a Group and Sub-groups within that group (- selecting the Group implies that all Sub-groups have potentially played a part).

Examples: The summary shown in Figure 12 was written using Groups of labels, including those shown in Figure 14. Here we show sentences in the summary (see paragraphs 1, 2 and 4), linked to certain Groups.

1. “Commenters also questioned how a fine would address congestion, the underlying cause of delays.”

   Link to Group: How is fining NR going to help with the problem of congestion/late running trains?

2. “Some pointed out that the train operating companies are also partly to blame for delays, but it was noted that there are separate penalties for them.”

   Link to Sub Group: TOCs also to blame (8 commenters)

3. “Many questioned the point of using the fine to improve Wifi service on trains.”

   Link to Group: Why use the fine on wifi? (9 commenters)

4. “Many suggested directors pay or bonuses should be cut instead; others suggested the fine could be part of a covert coalition plan to run the service down leading to performance issues which could be used as an argument for re-privatisation.”

   Link to Groups: NR directors bonuses (6 commenters); Public ownership status of NR (6 commenters) (see esp. Label 40)

5. “Finally, a group of commenters complained about NRs planned extra investment in the south-east commuter network arguing that it already received disproportionate resource.”

   Link to Group: Not fair that SE gets the investment (6 commenters)

(Labels created for comments sourced from: http://www.theguardian.com/business/2014/jul/07/network-rail-fined-50m-pounds-late-trains)

Group: (6 commenters) Public ownership status of NR
label for comment 33 [C2H4n – 3] :Suggests effect of fines will be for fares to go up, since NR will not want to pay from profits and will pass on to TOCs who will pass on to passengers.
label for comment 34 [Cynic24 – 3.1] :Points out that NR are a non-profit public body, so they have no profits.
label for comment 35 [C2H4n – 3.2] :Disagrees that NR is a non-profit public body – is a ”not for dividend” private company limited by guarantee
label for comment 36 [Cynic24 – 3.3] :Says effect of latter is the same as non-profit – ”not for dividend” private company done to keep debt off public accounts. Will change anyway in Sept to officially become a public body.
label for comment 38 [C2H4n – 3.5] :Wonders how NR becoming officially public squares with coalition view of private = good, public = bad.
label for comment 40 [Cynic24 – 3.7] :Office of Nat. Stats made decision to reclassify NR as public not for profit rather than ”not for dividend” private company limited by guarantee. Suggests NR being public suits coalition so that can grind it down leading to performance issues which could be used as an excuse for try to privatise again.

Group: How is fining NR going to help with the problem of congestion/late running trains?
label for comment 42 [Agir – 4] :If punctuality problem due to congestion how will fines help?
label for comment 90 [gristsparger – 12] :Why not fine the weather?

Sub Group: TOCs also to blame (8 commenters)
label for comment 20 [Deviant – 2.1] :How is fining NR going to help: agrees with previous. Suggest TOCS are to blame too; suggests disaster recovery planning.
label for comment 43 [MIAsin – 4.1] :Aren't TOCs responsible for punctuality?
label for comment 44 [FellOffMeChair – 4.2] :TOCs responsible: not if the problem is signalling, overrunning maintenance, etc.
...

Group: (6 commenters) Not fair that SE gets the investment
label for comment 50 [showmaster – 5] :SE gets the investment: complains most heavily subsidised rail users get even more subsidy. Contrasts with reneging on promises to fund electrification in Wales and suggest this is Cameron/Osborne vote buying and greed/selfishness of the SE in general.
label for comment 51 [Cynic24 – 5.1] :SE gets the investment: agrees – same with all gov't funding, arts, transport whatever. London and surrounds get disproportionate slice.
label for comment 52 [ricmondo – 5.2] :SE gets the investment: queries whether this last is true.
label for comment 53 [Cynic24 – 5.3] :SE gets the investment: asserts last is true and suggest searching web.
label for comment 54 [Orthus – 5.4] :SE gets the investment: agrees – take money from all rail travellers and give it to SE commuters.
label for comment 79 [ricmondo – 9.2] :Questions thread-heads swipe at SE. Why – SE has busiest trains/most crowded stations.

Group: (6 commenters) NR directors bonuses
label for comment 55 [RousselBland – 6] :NR directors' bonuses: Sarcastic remark about minding the gap between performance and directors' bonuses.
label for comment 56 [Skathi – 6.1] :NR directors' bonuses: Snide response about previous comment's humour. Notes directors' bonuses have been cut from 160% of salary to 20%
label for comment 57 [UncleMartin – 6.2] :NR directors' bonuses: 20% of salary still a large bonus
...

Group: (9 commenters) Why use the fine on wifi?
label for comment 61 [dunless – 7] :Use of fine on Wifi: better spent on addressing issues that made trains run late. Questions whether executive bonuses were affected as a result of the punctuality failure.
label for comment 77 [viper61 – 9] :Use of fine on Wifi: how will this speed up trains. Suggests won't benefit rail users as trains are too crowded to use equipment. Suggest main beneficiaries will be train operators/shareholders who are getting a public handout. Also suggests using fine this way sacrifices public safety so to address that charges will need to be increased by NR to operators who will pass on to public.
label for comment 78 [bachemobile – 9.1] :Use of fine on Wifi: sarcastic response saying you can carry on working when late (if you have a seat).
label for comment 79 [ricmondo – 9.2] :Questions thread-heads swipe at SE. Why – SE has busiest trains/most crowded stations.
...

Figure 14: Example showing labels organized into five 'Groups' and one 'Sub-group'