DEMO 2004/6

Linear Evolving Summarization: The First Results

Stergos D. Afantenos and Vangelis Karkaletsis
Software and Knowledge Engineering Laboratory
Institute of Informatics and Telecommunications,
National Centre for Scientific Research (NCSR) "Demokritos"
{stergos,vangelis}@iit.demokritos.gr

December 8, 2004

Contents

List of Figures

List of Tables

Abstract

We examine a method for the creation of multi-document *evolving* summaries, *i.e.* summaries of events that evolve through time. We distinguish such summaries into *linear* and *non-linear*. In this paper we focus in summaries that evolve linearly. In order to tackle the problem we introduce the notion of cross-document relations which connect some simple structures, which we call messages, in two exes: *synchronically* and *diachronically*. Apart from the theoretic background we provide, we also report on several Machine Learning experiments that we performed in order to automatically extract the messages and the synchronic and diachronic relations that connect them.

Chapter 1

Introduction

With the advent of the Internet, access to many sources of information, although possible in the past as well, has now become much more easier. One problem that arises though from this fact, is that of the information overflow. Imagine, for example, that someone wants to keep track of an event that is being described on various news sources, over the Internet, as it evolves through time. The problem is that there exist a plethora of news sources that it becomes very difficult for someone to compare the different versions of the story in each source. Furthermore, the Internet has made it possible now to have a rapid report of the news, almost immediately after they become available. Thus, in many situations it is extremely difficult to follow the rate with which the news are being reported.

In such cases, we suppose that a text summarizing the events from the various sources would be handy. In this paper we are concerned with the automatic creation of summaries from multiple documents which describe an event that evolves through time. Such a collection of documents usually contains news reports from various sources, each of which provides novel information on the event as it evolves through time. In many cases the sources will agree on the events that they report and in some others they will adopt a different viewpoint presenting a slightly different version of the events or possibly disagreeing with each other. Such a collection of documents can, for example, be the result of a Topic Detection and Tracking system (Allan et al. 1998).

Studying such a collection of documents, one will, inevitably, not fail to notice that those documents are related. In our study we have come to conclude that one can distinguish the relation between those documents into two categories: *Synchronic* relations and *Diachronic* relations. Loosely speaking, synchronic relations are mostly concerned with the agreement, disagreement, generalizations, etc, of certain subevents across documents which lie on the

same temporal horizon. Diachronic relations, on the other hand, are concerned with the evolution of a subevent through time.

Another distinction that we made through our study concerns the kinds of evolution that exist in the description of evolving events. In general, we can distinguish an evolution of the description of events into two categories: *linear evolution* and *non-linear evolution*. In the first category, linear evolution, we have descriptions of events that are coming in constant and possibly predictable quanta of time from many sources. This means that if the first news story q_0 comes at moment t_0, then we can assume that for each source the story q_n will come at time $t_n = t_0 + n * t$, where t is the constant amount of time that it takes for the news to appear. The second category, non-linear evolution, encompasses the rest of the cases. In this paper we will focus on the presentation of our research that concerns the summarization of events that evolve linearly.

Chapter 2

Motivation

An important aspect in Multi-document Summarization is the identification of similarities and differences between the documents (Mani 2001; Afantenos, Karkaletsis, and Stamatopoulos 2004). Mani and Bloedorn (1999), for example, identify similarities and differences among *pairs* of isolated documents by comparing the graphs that they derive from each document, which are based heavily on various lexical criteria. Our approach, in contrast, does not take into consideration isolated pairs of documents, but instead tries to identify the similarities and differences that exist between the documents in two directions: *synchronically* and *diachronically*.

What we mean by this is that as an event occurs, at a certain point of time, it is being described by various sources, with varying degrees of agreement, disagreement etc. Thus, in the synchronic level we are trying to identify the similarities and differences that exist between the various sources. On the diachronic level now, we have the description by each source of the evolution of that event. In that level we try again to identify similarities and differences, which now are focused on each source separately.

From a different perspective now, Radev (2000), inspired by Mann and Thompson's (1987) *Rhetorical Structure Theory*, proposed the Cross-document Structure Theory (CST) which tries to extend the rhetorical relations that exist within a document to multiple documents. The CST relations that he proposes are supposed to be domain independent and can hold among documents, paragraphs, sentences or phrases. In a later work on CST (Zhang, Blair-Goldensohn, and Radev 2002) perform some experiments with human subjects which reveal that, firstly, the judges do not believe that the CST relations exist between anything else but sentences, and, secondly, that the inter-judge agreement for the relations that hold between the sentences was very low.

Although we have some reservations concerning the domain independence

of the CST relations (Afantenos et al. 2004), we were nevertheless motivated by this approach, so we used some cross-document relations in order to capture the similarities and differences that exist between the documents. Our relations though are not domain independent, and they do not hold between documents or textual units but instead they hold between some simple structures which we call *messages*, and which are strongly related with the ontology of the domain.

In the following section, we will make more concrete and formal the notion of the messages and relations and we will present our methodology for the automatic creation of evolving summaries from multiple articles in section 4. The methodology will become clear in section 5 through a case study. In section 6 we present in more detail the related work. In section 7 we give the conclusions of this paper, and we present several ideas and ongoing work on *non-linear* evolving summarization.

Chapter 3

Some Definitions

The careful reader will have noticed that in the introduction of this paper, we have mentioned that the relations which exist between the documents, relate several subevents that are described in the documents. What we mean with this is that, in contrast with Radev's (2000) CST relations that connect textual units or documents,[1] we connect what is "represented" by the several textual units. Furthermore, we believe that a set, or taxonomy, of relations should not be general and domain independent, as Radev (2000) proposes, but instead they should be oriented more towards the pragmatics of each domain.[2] Before proceeding with the detailed presentation of our methodology, we would like to give a more formal definition to the messages and the relations.

3.1 Messages

In order to capture what is represented by several textual units, we employ some simple structures which we call *messages*. A message is composed from two parts: its *type* and a list of *arguments* which take their values from the domain *ontology*. In other words, a message can be defined as follows:

$$\texttt{message\_type (arg}_1\texttt{, ... , arg}_n\texttt{)}$$
$$\text{where } \texttt{arg}_i \in \text{Domain Ontology}$$

In some cases, a message definition will be accompanied by a set of *constraints* on the values that the arguments can take. Of course, messages lie inside

[1] In fact, only sentences have been examined so far (Zhang, Blair-Goldensohn, and Radev 2002; Zhang, Otterbacher, and Radev 2003; Zhang and Radev 2004).

[2] This does not mean that we do not believe that domain independent relations could not possibly exist. An example could be the relations agreement and disagreement, which can obviously be independent of domain.

documents, so we can say that they have associated with them information on the source of the document and the time that it was published. An example of a message definition will be given in the case study we present in section 5.

3.2 Relations

Relations, as we have already mentioned, can be of two types: *synchronic* and *diachronic*. If we represent a relation r as a pair of messages $\langle m_1, m_2 \rangle$, where m_1 and m_2 are two messages, then a relation will be synchronic iff

$$m_1.\texttt{time} = m_2.\texttt{time} \text{ and } m_1.\texttt{source} \neq m_2.\texttt{source}$$

and diachronic iff

$$m_1.\texttt{time} > m_2.\texttt{time} \text{ and } m_1.\texttt{source} \neq m_2.\texttt{source}$$

We have to note that a relation has a directionality. As is evident, diachronically a relation can hold from a past time to a future time. In the case of a synchronic relation (*e.g.* agreement) a relation can have both directions, in which case we have in fact two relations.

In order to define a relation in a domain we have to provide a *name* for it, and describe the conditions under which it will hold. The name of the relation is in fact *pragmatic* information, which we will be able to exploit later during the construction of the summary. The conditions that a relation holds are simply some rules which describe the *temporal distance* that two messages should have (0 for synchronic and more than 1 for diachronic) and the characteristics that the arguments of the messages should exhibit in order for the relation to hold. Thus, not all pairs $\langle m_1, m_2 \rangle$ of messages can hold a relation, but only the ones whose arguments exhibit particular characteristics as they have been specified.

Furthermore, it is crucial to note here the importance that time and source position have on the relations, apart from the values of the messages' arguments. Suppose, for example, that we have two identical messages. If the have the same temporal tag, but belong to different sources, then we have an *agreement* relation. If, on the other hand, they have the same source but chronological distance one, then we speak of a *stability* relation. Finally, if they have different sources and chronological distance more than two, then we have no relation at all. Thus we see that, apart from the characteristics that the arguments of a message pair $\langle m_1, m_2 \rangle$ should exhibit, the source and temporal distance also play a role for that pair to be characterized as a relation.

In section 5 we will give concrete examples of messages and relations for a particular case study. But before that, let us present a methodology for identifying and exploiting those messages and cross-document relations.

Chapter 4

The methodology

Before proceeding with the presentation of our methodology, we would like to put it in the general context of our multi-document summarization system. The system we have developed is a query-based summarization system, meaning that the summary it produces is an answer to a question that a user has posed. Before the user provides the query, the system has performed a preprocessing stage to the documents, during which it has identified all the messages and relations among them and has placed them in a structure which we call *grid* (see section 5 for more information on the grid). In order to identify the messages, our platform employs an Information Extraction (IE) subcomponent. Relations are identified according to the rules that accompany each one, during their specification. After the user has provided the query, the system identifies the various messages that are relevant to the query, as well as the relations that connect them. Thus, in essence the system *extracts a subgrid* from the original grid which is in fact the answer to the user query. This subgrid is passed to an Natural Language Generation (NLG) subcomponent which creates the final summary.

As is probably evident from the discussion thus far, our approach to multi-document evolving summarization does not claim any independence of domain. On the contrary, the methodology we will present, and its computational exploitation, are domain-dependent. This means that before creating a computational system, one has to have a certain amount of domain knowledge. This domain knowledge is, in essence, acquired by the *specification* of the messages and the relations. In order to reach a conclusion for the those specifications, we have developed a methodology which, for each domain, requires four steps. Those four steps, one can argue, are independent of domain, in the sense that one has to follow essentially the same path for each domain. On the other the exploitation of those specifications (*e.g.* the automatic extraction of messages from the corpora) depend on the do-

main, although even there we can provide some general ideas independent of domain.

In what follows, we present those steps, while on section 5 we will present the approach and algorithms we used in order to computationally exploit those specifications for a particular domain.

4.1 Corpus Collection

The first stage of the methodology is, of course, the collection of a corpus of documents which we would like to be summarized. Since the methodology we propose concerns multi-document summaries from events that evolve through time, the documents should contain descriptions of related events as they evolve through time, from multiple sources. In fact, an ideal input to our system would be a cluster of documents from a Topic Detection and Tracking system (Allan et al. 1998).

4.2 Creation of a domain ontology

The next step in our methodology involves the creation of an ontology. This step is an essential one, since we will use this ontology in the following step in order to specify the arguments of the messages.

4.3 Specification of the Messages

During this stage there are two tasks that we have to perform. The first one is to identify the message types, and the second is to provide their arguments as well as some possible constraints that should between the arguments. One can tackle both tasks together, or alternatively, reach firstly a conclusion on the types of messages that the domain contains and, later, on the arguments that those messages will have. As we have earlier mentioned, the arguments should take their values from the ontology that was created in the previous step, so for each argument, one should provide the ontology types that it can take, with probably some additional constraints on those types, which will reflect various pragmatic constraints of the domain.

The main idea behind this is to provide a level of abstraction, which will enable us to capture all the important subevents or types of information that one would have liked to see summarized, and thus ultimately ignore altogether the underlying text and deal with this abstract structure, or grid

as we call it. This abstraction will be later considerably enhanced with the addition of the relations as well.

4.4 Specification of the Relations

Once we have completed the specification of the messages, we should specify the cross-document relations that hold between the messages. The relations should be divided into two categories: *synchronic* and *diachronic*. These cross-document relations can be as general as the ones that the *Rhetorical Structure Theory* (Mann and Thompson 1987) or the Cross-document Structure Theory (CST) (Radev 2000) provides, or they could even be completely domain specific. We should always bare in mind that the relations express *pragmatic* information, which will later be used by the NLG system. Thus, the main purpose that the relations have, is to serve as better as possible the summarization task at hand.

The previous steps require some manual work which will allow us to capture the domain knowledge that we need for each domain. In the following section we are describing, through a particular case-study, how one can computationally identify the messages and the relations.

Chapter 5

A Case Study of Linear Evolution

Having provided a theoretical approach to the messages, relations and linear or non-linear evolution, it is time now to give some concrete examples, and provide a computational approach on how one can identify the messages and the relations.

We will do this through a case study for a particular domain. The domain we have chosen is that of the descriptions of football matches. In this domain, we have several events that evolve; the performance of a player or a team, for example, as the championship progresses. According to the definitions we have given, the evolution of this domain is *linear*. The reason for this is that we have a match each week which is then being described by several sources. Thus, for a particular team, if the first match q_0 was described in t_0, then each match q_n will be described in $t_n = t_0 + n * t$ where t is of course one week.[1] Thus we can say that the particular domain evolves in a linear manner.

As our methodology requires, in order to create multi-document evolving summaries, we have to provide some knowledge of the domain to the system. This knowledge is provided through the ontology and the specification of the messages and the relations, following the four steps described in the previous section. In the following subsections we describe how we provided this domain knowledge, and then how we computationally exploited it for the domain of the descriptions of football matches.

[1] Other domains that one can consider as having a linear evolution is the news from the stock market. Usually, the specialized news agencies provide information on the several stocks, after the stock market closes. In this case t would be a day.

5.1 Domain Knowledge

5.1.1 Corpus Collection

According to our methodology, the first thing that we need to do is collect some corpora on the domain. As we have earlier said, those corpora could ideally be a cluster of news from a Topic Detection and Tracking system, but in this case we opted for manual collection of the corpora. Thus, we collected descriptions of football matches, from various sources, for the period 2002-2003 of the Greek football championship. This championship contained 30 rounds. We focused on the evolution of a certain team, as it was described by three sources, so we had in total 90 documents describing the evolution of that team for the particular championship.

5.1.2 Ontology Creation

The following step, according to the suggestions of our methodology, is the creation of an ontology for the domain. Thus we proceeded towards this goal. The main entities that the created ontology included were persons, which we divided in players, coaches, referees, spectators, etc, some temporal concepts (minutes, first/second half, etc) and various other elements such as the cards. An excerpt of that ontology you can see in Figure 5.1.

```
Degree                Card
Person                  Yellow
  Referee               Red
  Assistant Referee   Team
  Linesman            Temporal Concept
  Coach                 Minute
  Player                Duration
  Spectators              First Half
    Viewers               Second Half
    Organized Fans        Delays
Round                     Whole Match
```

Figure 5.1: An excerpt from the domain ontology

5.1.3 Specification of the Messages

After the creation of the ontology, we proceeded with the specification of the messages. Since our system is a query-based one, we tried, during the process

of the specification of the messages, to concentrate on the most important events or facts on which users would, most possibly, pose questions. Furthermore, we tried to be focused on the fact that those messages should be depicting events that evolve through time, or some facts that a user would be interested in knowing. In the end of this process we concluded on the following set of 23 message types:

Absent, Behavior, Block, Card, Change, Comeback, Comment, Conditions, Expectations, Final_Score, Foul, Goal_Cancelation, Hope_For, Injured, Opportunity_Lost, Penalty, Performance, Refereeship, Satisfaction, Scorer, Successive_Victories, Superior, Win, System_Selection

Examples of full message specifications you can see in Figure 5.2.

performance (entity, in_what, time_span, value)

entity	:	player or team
in_what	:	Action Area
time_span	:	Minute or Duration
value	:	Degree

card (player, time_span, value)

player	:	Player
time_span	:	Minute or First Half, or Second Half or Whole Match or Delays
value	:	Card

Figure 5.2: Some message specifications

5.1.4 Specification of the Relations

The next step of the proposed methodology requires the specification of the cross-document relations. As we have mentioned those relations will be divided into two categories, synchronic and diachronic. For the football domain, we identified a total set of twelve cross-document relations, six on the synchronic and six on the diachronic axis. You can see those relations Table 5.1.

Since this was a pilot-study during which we examined mostly the viability of our methodology, we limited the study of the cross-document relations in relations that connect the *same* message types. Thus both the synchronic and the diachronic relations connect the same types, although further studies might reveal that different message types can be connected with some sort of relations. Furthermore, concerning the *diachronic* relations we limited our study in relations that have chronological distance only one.[2] In other words

[2]Chronological distance zero makes the relations synchronic.

Diachronic Relations	***Synchronic Relations***
– Positive Graduation	– Agreement
– Negative Graduation	– Near Agreement
– Stability	– Disagreement
– Repetition	– Elaboration
– Continuation	– Generalization
– Generalization	– Preciseness

Table 5.1: Synchronic and Diachronic Relations in the Football Domain

we did not examine if there exist relations for chronological distances more than one, and if they do how our current relations are affected. We provide more discussion on those issues in section 7.

Of course, as the methodology requires each relation type should be accompanied by its specifications, *i.e.* the conditions under which it holds. What this means is that one should provide some rules which, given two messages m_1 and m_2, explain the properties that those messages should have in order to have a relation $r\langle m_1, m_2\rangle$. For the relations that we have provided for this domain, the way to proceed towards the creation of the relations' specifications is for each one to provide the message types for which it can hold and the characteristics the arguments of those messages should have. Alternatively, what we did was for each message type we provided the conditions that their arguments should exhibit in order for those messages to be connected with certain relations. Examples of such specifications for the message type `performance` you can see in Figure 5.3.

A question that can arise is the following: *How does time affect the relations you create?* To answer that question, imagine having two identical messages, in different documents. If the documents have chronological distance zero, then we have an *agreement* relation. If the messages come from the same source but have chronological distance 1, then we have a *stability* relation. Finally, if the messages come from different sources and have chronological distance more than one, then we have no relation at all. Thus, indeed, time does affect the relations.

5.2 Computational Approach

As we have mentioned in the beginning of section 4, the summarization system that we have created, given the ontology of the domain and the specifications of the messages and relations, manages to create summaries

Performance
Assuming we have the following two messages:

performance (entity$_1$, in_what$_1$, time_span$_1$, value$_1$)
performance (entity$_2$, in_what$_2$, time_span$_2$, value$_2$)

Then the following Diachronic and Synchronic relations hold:

Diachronic Relations

- **Positive Graduation** iff
  (entity$_1$ = entity$_2$) and (value$_1$ < value$_2$)
- **Stability** iff
  (entity$_1$ = entity$_2$) and (value$_1$ = value$_2$)
- **Negative Graduation** iff
  (entity$_1$ = entity$_2$) and (value$_1$ > value$_2$)

Synchronic Relations

- **Agreement** iff
  (entity$_1$ = entity$_2$) and (value$_1$ = value$_2$)
- **Near Agreement** iff
  (entity$_1$ = entity$_2$) and (value$_1$ $\approx$ value$_2$)
- **Disagreement** iff
  (entity$_1$ = entity$_2$) and (value$_1$ $\neq$ value$_2$)

Figure 5.3: Specifications of Relations

which are answers to questions that concern the evolution of several subevents in the domain, *e.g.* the performance of a player or a team. In order to do that we have to extract the messages with their arguments, and the relations that connect them, and subsequently organize them in a structure which we call a *grid* (see Figure 5.4). This grid reflects exactly the fact that the domain that we have used in this case study exhibits linear evolution, in the sense that was described in the introduction of this paper. If we take a horizontal "slice" of the grid, then we will have descriptions of events from all the sources, for a particular time unit. If, on the other hand, we take a vertical "slice" of the grid, then we have the description of the evolution of an event from a particular source.

In this section we will provide the computational means that we used in order to automatically identify the messages and their arguments in the documents and the relations that hold between them, in order to construct

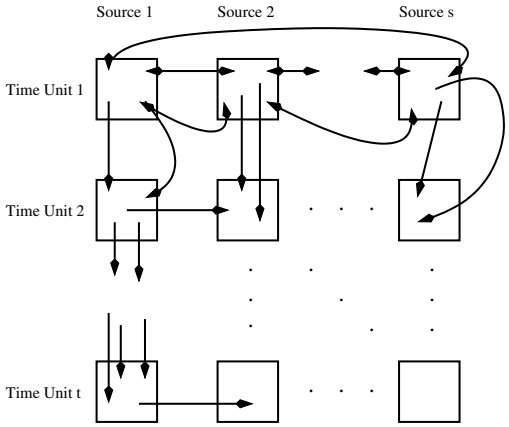

Figure 5.4: The Grid structure with Synchronic and Diachronic Relations

the structure of a grid.

5.2.1 Messages Extraction

The subsystem that we have used in order to extract the messages and their arguments is depicted in Figure 5.5. At this point, the reader has to bear in mind that this paper presents a pilot-study whose aim was to examine the viability of the proposed method of tackling the problem of evolving summarization. Thus, we have not used a very sophisticated approach on the problem of message extraction, Nevertheless, the results were more than acceptable, as we will soon present.

Figure 5.5: The message extraction subsystem

Preprocessing.

The message extraction subsystem, which was developed under the *Ellogon* platform (Petasis et al. 2002) is composed from a preprocessing stage, which includes tokenization and sentence splitting. Next comes the Named Entity Recognition and Classification (NERC) stage during which we try to identify the Named Entities (NEs) in the documents and classify into the categories

that the ontology proposes. The method we used was heavily based on lists of NEs.

The next two steps are the core of the message extraction subsystem. As you can see, we divided the problem of the extraction of the messages into two separate subproblems. In the first we try to identify the *types of messages* that exist in the documents, while in the second we try to extract their arguments.

Message Classification.

Concerning the identification of the message types, we approached it as a classification problem. From a study that we carried, we concluded that in most of the cases the mapping from sentences to messages was one-to-one, *i.e.* in most of the cases one sentence corresponded to one message. Of course, there were cases in which one message was spanning more than one sentence, or that one sentence was containing more than one message. Nevertheless, since our approach was meant to simply be an initial pilot-study, we concluded that we would tackle this problem as a classification problem with a one-to-one mapping from messages to sentences.

Thus we used a bag-of-words approach according to which we represented each sentence as a vector from which the stop-words and the words with low frequencies (four or less) were removed. We performed the experiments both with the words *stemmed* and *unstemmed*. Furthermore we added into those two sets of vectors information concerning the number of times that each *named entity category* appears in the sentence. Of course, in order to perform the training phase of the experiments, in each of the vectors we appended the *class* of the sentence, *i.e.* the type of message; in case a sentence did not corresponded to a message we labeled that vector as belonging to the class *None*. This resulted into *four* series of vectors and corresponding experiments that we performed.

In order to perform the classification experiments we used the WEKA platform (Witten and Frank 2000). The Machine Learning algorithms that we used where three: *Naïve Bayes, LogitBoost* and *SMO*. For the last two algorithms, apart from the default configuration, we performed some more experiments concerning several of their arguments. Thus for the LogiBoost we experimented with the number of iterations that the algorithm performs and for the SMO we experimented with the complexity constant, with the exponent for the polynomial kernel and with the gamma for the RBF kernel. For each of the above combinations we performed a *ten-fold cross-validation* with the annotated corpora that we had. The results of the above experiments are presented in Table 5.2.

Taking a look at that table there are several remarks that we can make. Firstly, the LogitBoost and the SMO classifiers that we used outperformed, in all the cases, the Naïve Bayes which was our baseline classifier. Secondly, the inclusion of the NE types in the vectors gave a considerable enhancement to the performance of all the classifiers. This is rather logical, since almost all the messages contain in their arguments NEs. The third remark, concerns the stemmed and the unstemmed results. As we can see from the table, the algorithms that used vectors which contained unstemmed words outperformed the corresponding algorithms which used vectors whose words had been stemmed. This is rather counterintuitive, since in most of the cases using stemming one has better results.

Ultimately, the algorithm that gave the best results, in the experiments we performed, was the SMO with the default configuration for the unstemmed vectors which included information on the NE types. Thus, we integrated this trained classifier in the message extraction subsystem, which you can see in Figure 5.5. In section 7 we have a discussion concerning whether we could perform some more experiments which would yield better results.

Argument Extraction

The final step for the message extraction subsystem is the filling in of the arguments. In order to perform that we employed several heuristic methods which are almost completely domain-specific.

As we noted above, one of the drawbacks of the classification approach that we used is that there are some cases in which we do not have a one-to-one mapping from sentences to messages, *i.e.* there some cases in which, for example, one sentence can correspond to many messages. During this phase of the message extraction subsystem we managed with heuristics to remedy for many of the cases.

In Table 5.3 we show the final performance of the subsystem as a whole, when compared against manually annotated messages on the corpora used. As you can see from that table although the vast majority of the messages that we automatically extract are correct, we do not manage to extract all the messages. This is reflected by the high percentage of precision and the not so high results of recall. The combination of those measures gives as an F-measure around 70% which one could argue is quite satisfying.

5.2.2 Extraction of Relations

As is evident from Figure 5.3, once we have identified the messages in each document and we have placed them in the appropriate position in the grid,

then it is fairly straightforward, through their specifications, to identify the cross-document relations among the messages.

In order to achieve that we implemented, entirely from scratch, a platform which was written in Java. This platform takes as input the extracted messages with their arguments from the previous subsystem and it is responsible for the incorporation of the ontology, the representation of the messages in an abstract form and the extraction of the synchronic and diachronic cross-document relations. Ultimately, through this platform we manage to represent the *grid*, which carries an essential role for our summarization approach.

The reason for this is that since our approach is a query based one, we would like to be able to pose queries and get the answers from the grid. The platform that we have created implements the API through which one can pose queries to the grid, as well as the mechanism that extracts from the whole grid structure the appropriate messages and the relations that accompany them, which form an answer to the question. Those extracted messages and relations form a sub-grid which can then be passed to an NLG system for the final creation of the summary.

As of writing this paper, everything has been implemented except the mechanism that transforms the natural language queries to the API that will extract the sub-grid. Additionally, we do not have a connection with an NLG system, but instead we have implemented some simple frame-based (or canned text) mechanism that transforms the sub-grid to natural language.

Classifier		**Correctly Classified Instances**
Without NE types		
stemmed	Naïve Bayes	60.6693 %
	LogitBoost default	72.7443 %
	LogitBoost $I = 5$	71.8876 %
	LogitBoost $I = 15$	72.2892 %
	SMO default	73.6011 %
	SMO $C = 0.5$ $E = 0.5$ $G = 0.001$	68.9692 %
	SMO $C = 1.5$ $E = 1.5$ $G = 0.1$	74.4578 %
unstemmed	Naïve Bayes	62.2758 %
	LogitBoost default	75.8768 %
	LogitBoost $I = 5$	74.9398 %
	LogitBoost $I = 15$	76.6533 %
	SMO default	79.2503 %
	SMO $C = 0.5$ $E = 0.5$ $G = 0.001$	75.2343 %
	SMO $C = 1.5$ $E = 1.5$ $G = 0.1$	77.992 %
Including NE types		
stemmed	Naïve Bayes	63.8286 %
	LogitBoost default	78.0991 %
	LogitBoost $I = 5$	76.1981 %
	LogitBoost $I = 15$	78.2062 %
	SMO default	75.9839 %
	SMO $C = 0.5$ $E = 0.5$ $G = 0.001$	72.5301 %
	SMO $C = 1.5$ $E = 1.5$ $G = 0.1$	75.7965 %
unstemmed	Naïve Bayes	64.2035 %
	LogitBoost default	78.9023 %
	LogitBoost $I = 5$	77.4565 %
	LogitBoost $I = 15$	79.4645 %
	SMO default	79.6252 %
	SMO $C = 0.5$ $E = 0.5$ $G = 0.001$	76.8675 %
	SMO $C = 1.5$ $E = 1.5$ $G = 0.1$	78.5007 %

Table 5.2: The results from the classification experiments

Precision	:	92.3745
Recall	:	55.1585
F-Measure	:	69.0725

Table 5.3: Recall, Precision and F-Measure on the grid

Chapter 6

Related Work

The work that we present in this paper is concerned with what we call multi-document *evolving summarization*, *i.e.* summarization of events that evolve through time. Of course, we are not the first to incorporate directly, or indirectly, the notion of time in our approaches to summarization. Lehnert (1981), for example, attempts to provide a theory for what she calls *narrative summarization*. Her approach is based on the notion of "plot units", which connect *mental states* with several relations, and are combined into very complex patterns. This approach is, in contrast with ours, single document. Unfortunately, she does not provide any computational implementation of her theory. Recently, Mani (2004) attempts to revive this theory of narrative summarization, although he also does not provide any concrete computational approach for its implementation.

From a different viewpoint, Allan, Gupta, and Khandelwal (2001) attempt what they call *temporal summarization*. In order to achieve that, they take the results from a Topic Detection and Tracking system for an event, and they put all the sentences one after the other in a chronological order, regardless of the document that it belonged, creating a stream of sentences. Then they apply two statistical measures *usefulness* and *novelty* to each ordered sentence. The aim is to extract those sentences which have a score over a certain threshold. This approach differs greatly from ours, since we do not have an extractive system, but an *abstractive* one. Furthermore, they are concerned with only one source of information, while we try to incorporate in our system the different viewpoints that the various sources might have, and present them to the user.

As we have said, our work requires some domain knowledge which is expressed through the ontology, and the messages' and relations' specifications. A system which is based also on domain knowledge is SUMMONS (Radev and McKeown 1998; Radev 1999). The main domain knowledge for this sys-

tem comes from the specifications of the MUC conferences (`citation`). This system takes as input several MUC templates and, applying a series of operators, it tries to create a baseline summary, which is then enhanced by various named entity descriptions collected from the Internet. The difference with our approach is that we do not rely in such complex structures as the MUC templates, but in much more simple ones. Furthermore, the operators that SUMMONS uses *alter* the templates by combining, eliminating, etc, several of their slots, whilst in our case we simply report the similarities, differences and the evolution by several relations, which do not alter the messages. Finally our system is a query-based one.

Concerning now the use of relations, there have been several attempts in the past which try to incorporate relations, in one form or another, for the creation of a summary. Salton et al. (1997), for example, try to extract paragraphs from a single document by representing them as vectors and assigning a relation between the vectors if their similarity exceeds a certain threshold. They then present various heuristics for the extraction of the best paragraphs.

Finally, Radev (2000) proposed the Cross-document Structure Theory (CST) which incorporated a set of 24 domain independent relations that exist between various textual units across documents. In a later paper Zhang, Blair-Goldensohn, and Radev (2002) reduces that set into 17 relations and perform some experiments with human judges. Those experiments reveal several interesting results. For example, human judges annotate only sentences, ignoring the other textual units (phrases, paragraphs, documents) that the theory suggests. Additionally, we see a rather small inter-judge agreement concerning the type of relation that connects two sentences. Nevertheless, Zhang, Otterbacher, and Radev (2003) and Zhang and Radev (2004) continue with some experiments, during which they try to use some Machine Learning algorithms in order to identify the cross-document relations. The algorithm they use is Boosting, and the results for each class of relation, using F-Measure, vary from 0.0513 to 0.4324. We have to note that, in contrast with us, in their approach (Zhang, Blair-Goldensohn, and Radev 2002; Zhang, Otterbacher, and Radev 2003; Zhang and Radev 2004) they still have not yet provided any implementation for the exploitation of the cross-document relations for the production of a summary.

Chapter 7

Conclusions and Future Work

The aim of this paper was to present our approach to the problem of *multi-document evolving summarization*, which we divide into linear and non-linear. In order to tackle the problem, we introduced some cross-document relations which depict the evolution of the subevents in two axes: *synchronic* and *diachronic*. Those relations connect some structures which we call messages, which reflect the main sub-events of the domain, and which heavily depend on the domain ontology.

Apart from presenting a methodology through which one can proceed for the specifications of the messages and the relations, we presented as well, through a case study, a computational system which is able to extract the messages and the relations for a particular domain. Furthermore, we presented a platform which is able to represent those messages and relations in structure which we call a *grid*, and which can be queried for a question, yielding a sub-grid which reflects the answer to the question. This sub-grid can later be transformed into natural language, through an NLG system.

Currently we are working on a more complicated domain, namely that of terroristic and non-terroristic hostages, whose evolution, according to the specification that we gave in the introduction of this paper, can be characterized as non-linear. The main challenges in non-linear evolution concern the synchronic relations. In the case of linear evolution it was quite clear what means synchronic, since the time units were discreet (see Figure 5.4), something which does not necessarily happen in the case of non-linear evolution.

Other issues that we currently explore concern the diachronic relations. More specifically, we investigate in more depth the role that time has on the relations and whether we can have diachronic relations of depth more than one. We also investigate whether we can have relations which connect different message types.

Concerning now the classification experiments, after the initial pilot-

study, we are certain that they can be much better, and we are actively working towards this goal. For example, as we have seen, almost all of the sentences that correspond to messages contain NEs, which means that we can ignore from the classification experiments all of the sentences which do not contain NEs (effectively, removing the *None* class that we had). Furthermore we examine means, other than heuristics, for the extraction of the arguments.

Bibliography

Afantenos, Stergos D., Irene Doura, Eleni Kapellou, and Vangelis Karkaletsis. 2004, May. "Exploiting Cross-Document Relations for Multi-Document Evolving Summarization." Edited by G. A. Vouros and T. Panayiotopoulos, *Methods and Applications of Artificial Intelligence: Third Hellenic Conference on AI, SETN 2004*, Volume 3025 of *Lecture Notes in Computer Science*. Samos, Greece: Springer-Verlag Heidelberg, 410–419.

Afantenos, Stergos D., Vangelis Karkaletsis, and Panagiotis Stamatopoulos. 2004. "Summarization from Medical Documents: A Survey." *Journal of Artificial Intelligence in Medicine*. In press.

Allan, James, Jaime Carbonell, George Doddington, Jonathan Yamron, and Yiming Yang. 1998, February. "Topic Detection and Tracking Pilot Study: Final Report." *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*. 194–218.

Allan, James, Rahuk Gupta, and Vikas Khandelwal. 2001. "Temporal Summaries of News Stories." *Proceedings of the ACM SIGIR 2001 Conference*. 10–18.

Lehnert, Wendy G. 1981. "Plot Units: A Narrative Summarization Strategy." In *Strategies for Natural Language Processing*, edited by W. G. Lehnert and M. H. Ringle, 223–244. Hillsdale, New Jersey: Erlbaum. Also in (Mani and Maybury 1999),.

Mani, Inderjeet. 2001. *Automatic Summarization*. Volume 3 of *Natural Language Processing*. Amsterdam/Philadelphia: John Benjamins Publishing Company.

———. 2004. "Narrative Summarization." *Journal Traitement Automatique des Langues (TAL): Special issue on "Le résumé automatique de texte: solutions et perspectives"* 45, no. 1 (Fall).

Mani, Inderjeet, and Eric Bloedorn. 1999. "Summarizing Similarities and

Differences Among Related Documents." *Information Retrieval* 1 (1): 1–23. Also in (Mani and Maybury 1999).

Mani, Inderjeet, and Mark T. Maybury, eds. 1999. *Advances in Automatic Text Summarization*. The MIT Press.

Mann, W. C., and S. A. Thompson. 1987. "Rhetorical Structure Theory: A Framework for the Analysis of Texts." Technical Report ISI/RS-87-185, Information Sciences Institute, Marina del Rey, California.

Petasis, George, Vangelis Karkaletsis, George Paliouras, Ion Androutsopoulos, and Costas D. Spyropoulos. 2002, May. "Ellogon: A New Text Engineering Platform." *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC 2002)*. Las Palmas, Canary Islands, Spain, 72–78.

Radev, Dragomir R. 1999. "Generating Natural Language Summaries from Multiple On-Line Sources: Language Reuse and Regeneration." Ph.D. diss., Columbia University.

———. 2000, October. "A Common Theory of Information Fusion from Multiple Text Sources, Step One: Cross-Document Structure." *Proceedings of the 1st ACL SIGDIAL Workshop on Discourse and Dialogue*. Hong Kong.

Radev, Dragomir R., and Kathleen R. McKeown. 1998. "Generating natural language summaries from multiple on-line sources." *Computational Linguistics* 24 (3): 469–500 (September).

Salton, Gerald, Amit Singhal, Mandar Mitra, and Chris Buckley. 1997. "Automatic Text Structuring and Summarization." *Information Processing and Management* 33 (2): 193–207. Also in (Mani and Maybury 1999).

Witten, Ian H., and Eibe Frank. 2000. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. San Francisco: Morgan Kaufmann.

Zhang, Zhu, Sasha Blair-Goldensohn, and Dragomir Radev. 2002, August. "Towards CST-Enhanced Summarization." *Proceedings of AAAI-2002*.

Zhang, Zhu, Jahna Otterbacher, and Dragomir Radev. 2003, November. "Learning cross-document structural relationships using boosting." *Proccedings of the Twelfth International Conference on Information and Knowledge Management CIKM 2003*. New Orleans, Louisiana, USA, 124–130.

Zhang, Zhu, and Dragomir Radev. 2004, March. "Learning Cross-document Structural Relationships using Both Labeled and Unlabeled Data." *Proceedings of IJC-NLP 2004*. Hainan Island, China.