Universität Karlsruhe (TH)
Forschungsuniversität • gegründet 1825

UNIVERSITÉ Concordia UNIVERSITY

Diplomarbeit

Automatic Analysis and Reasoning on Reported Speech in Newspaper Articles

Automatische Analyse und Verarbeitung von Direkter und Indirekter Rede in Zeitungsartikeln

cand. inform.
Ralf Krestel

Institut für Programmstrukturen
und Datenorganisation (IPD)
Fakultät für Informatik
Universität Karlsruhe,
Deutschland

CLaC Laboratory
Department of Computer Science
and Software Engineering
Concordia University,
Montréal, Canada

Tag der Anmeldung: 1. Juni 2006
Tag der Abgabe: 31. Januar 2007

Verantwortlicher Betreuer:
Prof. em. Dr.-Ing. Dr. h.c.
Peter C. Lockemann

Betreuender Mitarbeiter:
Dr.-Ing. René Witte

Externe Betreuerin:
Prof. Dr. Sabine Bergler

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Karlsruhe, den 31. Januar 2007

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis is concerned with extracting statements from newspaper articles and building up a belief representation system simulating a human believer. In this first chapter we will give a short motivation for developing such a system followed by an overview of the system and the structure of the thesis.

1.1 Motivation

Due to the huge success of the Internet, the Natural Language Processing (NLP) research community has developed whole branches that deal explicitly with vast amounts of information encoded in written natural language[1]. One goal is to gain knowledge about hard facts like "The number of inhabitants of city X" or the "Name of the president of country X". But a lot of information, especially within newspaper articles, are not hard facts that could be easily proven right or wrong. Often newspaper articles contain different views of the same event, or state controversial opinions about a certain topic. In this case the notion of *belief* becomes relevant.

For humans this is an everyday task, sometimes a conscious act, sometimes unconsciously adopted. Depending on context information and background knowledge, together with other belief structures, humans tend to believe certain statements while other statements get rejected. Although everybody uses the term *belief*, the definition is rather vague, and the processes taking place inside the brain while "someone is believing something" are not understood. The process of believing also varies between different humans, not only depending on different background knowledge but on different attitudes towards a coherent worldview or importance and ability of *logic* thinking.

A computational system, whose task should be to simulate a human newspaper reader by imitating his belief processing, has not only to take into account the extraction of believes stated in an article, but also the existing beliefs within the system. Such an artificial believer[2] should also be able to distinguish between different belief strategies, modelling the different human approaches. The problem of ambiguity inherent to natural language, and the lack of "real" natural language understanding of computational systems, enforces the use of methods limiting the correctness of the output to a probability.

The huge amount of data available for example in the Internet or special data bases represents a perfect ground for the use of such an artificial believer. The area of application is large. Potential users of an artificial believer system are for example:

- Companies interested in customers' opinions about their own products or products from a competitor.

[1] for example *Information Extraction*, *Summarization*, or *Information Retrieval*
[2] this term was shaped by [BW91]

- Governments interested in the opinions of people about their country or the goverment's work.
- Individuals, who wish to have a personalized news digest compiled automatically.

Our system was especially designed for the last group of users, but is not conceptually limited to this dedicated field of application. The next section presents an overview of our approach using fuzzy theory to build an artificial believer.

1.2 Fuzzy Believer

Our Fuzzy Believer application should model a human newspaper reader who develops his own point of view for current events described in newspaper articles. More specifically, we only rely on information stated within the grammatical construct of *reported speech*. This allows a clear assignment of statements to sources and enables the system to judge according to different degrees of reliability in a source.

In Figure 1.1 an excerpt of a typical newspaper article is presented with different views and opinions expressed using reported speech. Even in this short excerpt, opposing opinions can be found compelling the reader to take either one side or to believe neither (compare the last two sentences). The Fuzzy Believer is designed to make these decisions based on different strategies and presents a set of held beliefs and rejected beliefs after processing newspaper articles. The strategies used to model different human behavior are:

- Believe everything,
- Believe old news,
- Believe new news,
- Believe majority,
- Believe certain newspaper/reporter/source,
- Believe weighted majority.

The different strategies make it necessary to identify the topics statements deal with. And the Fuzzy Believer has to identify the polarity of the statements to detect opposite opinions.

Extracting reported speech from newspaper articles is the first step, delivering the raw material to work with in the next steps. These steps comprise processing the extracted statements. A two layer approach is solving the topic and polarity identification task. First, the system decides which statements are related and should therefore be considered to deal with the same topic. And secondly, the system identifies the *polarity or orientation* of statements for each topic. The output of the Fuzzy Believer is a set of statements that the system "believes". This set of beliefs can be computed according to different strategies according to different preferences.

1.2.1 Problem Definition

The task of building an artificial believer can be divided into different subtasks that can be solved separately. The first question is what kind of input to use to

North, Meese Often Discussed Contras, Hostage Release Efforts, Sources Say *By John Walcott*
Justice Department officials have said that Mr. Meese and Col. North regularly met or talked by telephone. Adm. Poindexter dismissed the warnings, saying administration lawyers had reviewed Col. North's activities in support of the rebels and had concluded that the Contra aid program was legal, the sources said. Separately, Reagan administration sources said, some White House officials became alarmed that Col. North apparently had been collecting and disbursing large amounts of cash for the Contras from his office on the third floor of the Old Executive Office Building next to the White House. The sources said Col. North had none of the required government vouchers for the money. Nevertheless, Reagan administration sources said Adm. Poindexter resisted repeated efforts to force Col. North to share responsibility for his secret Iran and Contra operations with other NSC staff members. During the spring of 1986, the sources said, NSC specialists in Central American affairs and in intelligence matters separately became concerned about some of Col. North's activities and asked to be included in overseeing them. Adm. Poindexter rejected the requests, sources said, and turned down a suggestion from another aide that the NSC staff's Middle East expert, Dennis Ross, be consulted about the Reagan administration's secret Iran policy. "It was clear that Ollie (Col. North) had someone's hand on his shoulder," said one Reagan administration official. "He was never perceived as an unauthorized loner." Administration officials said they believe Col. North also kept Vice President Bush and his national security adviser, Donald Gregg, informed about some aspects of the Contra aid network. Mr. Bush and his aides have tried hard to distance themselves from the scandal, but the team headed by independent counsel Lawrence Walsh has indicated it intends to delve into contacts that Mr. Bush's office had with Contra supporters, according to law enforcement officials. Interviews with some of Mr. Bush's aides were conducted after the Federal Bureau of Investigation began a full-scale criminal investigation into the matter at the beginning of December, according to law enforcement officials. White House officials said Mr. Bush was interviewed by FBI agents for 25 minutes on Dec. 12. Other officials said they believed a wide-ranging interview with Mr. Bush hasn't been scheduled.

Figure 1.1: Excerpts of a Wall Street Journal (WSJ) newspaper article from 01.23.87 dealing with the Iran-Contra Affair

let the system choose what to believe. The goal is to model a human newspaper reader and therefore the input texts of choice are newspaper articles. They are available through databases or the Internet and because of their occurrences in vast numbers a computational system to process them automatically is desirable. We further limited our field of application to reported speech within newspaper articles. This has the advantage that we can attribute content to a dedicated source and that the uttered statements are usually about a certain topic and represent a dedicated opinion about this topic. The additional information we can gain from reported speech apart from the content of the uttered statement has been analysed by Bergler [Ber92], and enriches the amount of information we can extract from newspaper articles like confidence of the reporter into the utterance. To make this information accessible, we have to identify and extract the different elements of reported speech. As can be seen in Figure 1.1, reported speech structures consist of up to four functional elements:

- A source of a statement,
- A reporting verb,
- A reported clause – the content of the utterance, and
- Optionally circumstantial information.

These elements have to be identified by the Fuzzy Believer and extracted. Another important aspect is the identification of same sources to ascribe the utterances to. In our example, different “officials” are mentioned and the system has to find out which expressions are referring to the same entity. The gained information has to be stored in a suitable representation, grouping statements of the same source together, and preparing the extracted sentences for further processing.

The topics of the statements have to be identified as a basic step to enable the system to chose between opposing opinions about a particular topic. In the example above, the system has to identify the topic “Mr. Bush was interviewed” and detect affirmative and contradicting statements. The key concept we use to identify, compare, and contrast the single statements with each other is fuzzy set theory, as described by Witte [Wit02a]. Therefore we have to find a suitable fuzzy representation of the statements and a way to process them.

1.2.2 Architecture

A general overview of the system’s architecture is shown in Figure 1.2. A component based architecture is suitable for NLP systems in general and for an artificial believer in particular.

Therefore the system is implemented within the GATE [CMBT02] framework as different components. It uses existing components shipped with GATE to preprocess the texts and to fulfill basic NLP subtasks. The components designed and developed for our system are:

- Reported Verb Finder,
- Reported Speech Finder,
- Profile Creator,
- Parser Wrapper / Predicat-Argument Structure Extractor, and
- Fuzzy Believer.

The single components are either implemented in Java or in JAPE (Section 5.1.1.2).

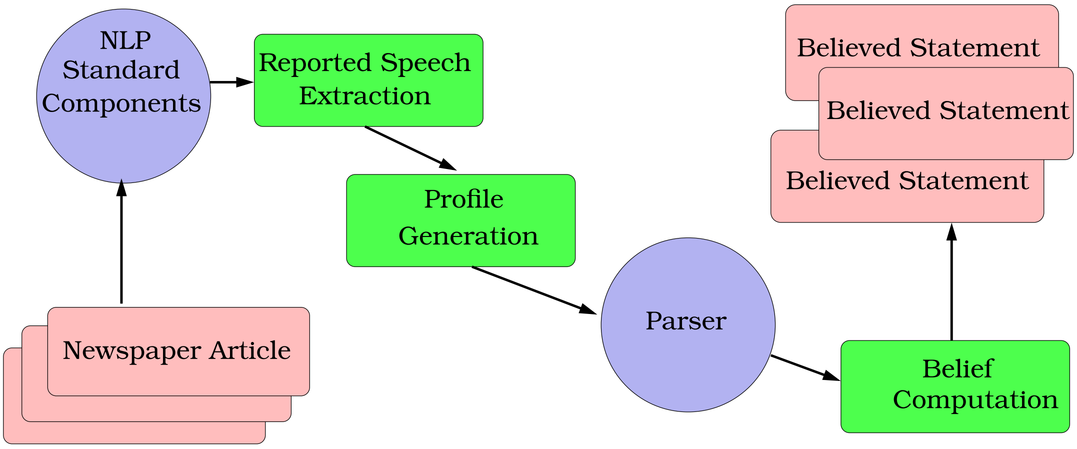


Figure 1.2: Fuzzy Believer system architecture overview

1.2.3 System Usage Example

The Fuzzy Believer was designed to process collections of newspaper articles. A typical scenario would be the processing of articles about one event or a series of related events. As source for the input of the system, news articles published in the Internet can be taken. Especially comparing articles with different publishing dates is interesting, as it often reflecting some kind of development of the reported story. New information is added or old, wrong information becomes substituted, or another actor is adding statements. This is the setting the Fuzzy Believer was designed for. It outputs held beliefs after processing the various articles according to different strategies. For the example above, which consists of only one article, it is hard to find opposing opinions and to prefer one because there is not enough information available to support either side. But if we add other newspaper articles dealing with the Iran-Contra Affair, the system will find opinions supported by a majority or can chose to believe statements that are newer and reject old contradicting ones. The result and output of the Fuzzy Believer are held beliefs and rejected beliefs in the shape of statements. Figure 1.3 shows a small sample output of our system.

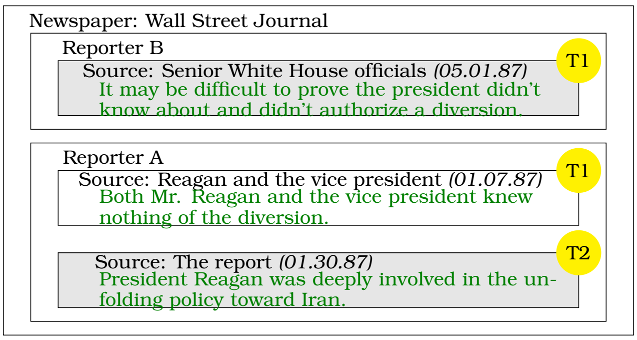


Figure 1.3: System's beliefs based on majority strategy

1.3 Structure of the Thesis

The thesis is divided into seven chapters. This Chapter 1 started with a short motivation for the thesis, followed by the definition of the task to be solved and a description of the Fuzzy Believer system. This description presents an example on how to use the system and gives an overview of the crucial features. The first chapter finishes with the description of the architecture and a preview of the obtained results.

Chapter 2 is dedicated to an overview of the NLP research area and existing, similar systems. The overview of the field first shows the development of NLP systems using different components and second presents some typical NLP applications. The chapter proceeds with the presentation of work related to the Fuzzy Believer. To conclude this chapter, we discuss the relation of our system with respect to the NLP area in general and in comparison to existing systems.

Chapter 3, "Theory and Foundations", discusses principal approaches used in building our Fuzzy Believer. In the first section of this chapter, the topic is "Reported Speech". Fundamental concepts and philosophical implications of "Belief" is the next major part. The theory of fuzzy systems concludes this chapter.

Presenting the "Design" of the Fuzzy Believer is the topic of Chapter 4. After giving an overview of the system design in total, the individual Fuzzy Believer components are described in more detail:

- The Reported Speech extraction,
- The Profile generation,
- Belief Representation, and
- The Fuzzy Believer.

The different fuzzy belief strategies close this chapter.

In Chapter 5, the implementation of the system is described. Starting with the GATE framework, which offers low level NLP components and an environment for our Fuzzy Believer components, we give an overview of the deployed components. The parsers we use are described in the following section; afterwards, the implementation of our Fuzzy Believer components is presented in detail. The actual components developed based on the design requirements:

- Reported Verb Finder,
- Reported Speech Finder,
- Profile Creator,
- Parser Wrapper / PAS Extractor, and
- Fuzzy Believer.

Chapter 6 contains the evaluation of the system. After introducing different basic concepts for evaluating NLP systems, the features that are evaluated are described. The corpora used for evaluation purposes are presented followed by the obtained results of our system. This also includes a "real world" system usage example. The last part in this chapter is the discussion of the results and the evaluation methods.

The last Chapter 7 is dedicated to a summary of the work, an overview of the results, and a preview of future work. This last part shows possible applications as well as improvements needed for better system performance.

Chapter 2

Literature Review and Related Work

We start by giving an overview and definition of Natural Language Processing and how our work can be classified in this area.

2.1 Overview of the Field

Natural Language Processing (NLP) can be seen as a sub-field of Artificial Intelligence (AI) and Computational Linguistics (CL). One of the many different definitions categorizes NLP as [Lid01]:

> "... a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications."

To make this definition more clear, [Lid01] also explains the used terms more elaborately:

- The imprecise notion of *range of computational techniques* is necessary because there are multiple methods or techniques that can be applied to accomplish a particular type of language analysis.

- *Naturally occurring texts* can be of any language, mode, genre, etc. The texts can be oral or written. The only constrain for the texts are to be in a language used by humans to communicate to one another.

- The notion *levels of linguistic analysis* refers to the fact that there are multiple types of language processing known to be at work when humans produce or comprehend language. It is thought that humans normally utilize all of these levels since each level conveys different types of meaning. But various NLP systems utilize different levels, or combinations of levels of linguistic analysis, and this is seen in the differences amongst various NLP applications.

- *To accomplish human-like language processing* reveals that NLP is considered a discipline within Artificial Intelligence (AI). And while the full lineage of NLP does depend on a number of other disciplines, since NLP strives for human-like performance, it is appropriate to consider it an AI discipline.

- *For a range of tasks or applications* points out that NLP is not usually considered a goal in and of itself, except perhaps for AI researchers. For others, NLP is the means for accomplishing a particular task. Therefore, you have Information Retrieval (IR) systems that utilize NLP, as well as Machine Translation (MT), Question-Answering, etc.

- *accomplish human-like language processing* is the goal of NLP as stated above. The choice of the word 'processing' is very deliberate, and should not be replaced with 'understanding'. For although the field of NLP was originally referred to as Natural Language Understanding (NLU) in the early days of AI, it is well agreed today that while the goal of NLP is true NLU, that goal has not yet been accomplished. A full NLU System would be able to:
  (i) Paraphrase an input text
  (ii) Translate the text into another language
  (iii) Answer questions about the contents of the text
  (iv) Draw inferences from the text

  While NLP has made serious inroads into accomplishing goals 1 to 3, the fact that NLP systems cannot, of themselves, draw inferences from text, NLU still remains the goal of NLP. There are more practical goals for NLP, many related to the particular application for which it is being utilized. For example, an NLP-based IR system has the goal of providing more precise, complete information in response to a user's real information need. The goal of the NLP system here is to represent the true meaning and intent of the user's query, which can be expressed as naturally in everyday language as if they were speaking to a reference librarian. Also, the contents of the documents that are being searched will be represented at all their levels of meaning so that a true match between need and response can be found, no matter how either are expressed in their surface form.

As we see, there is a lot of work to be done to get a working NLP system, especially if a system tries to draw inferences from texts like our Fuzzy Believer is supposed to do. But before we take a deeper look into the variety of NLP systems, we take a glimpse onto the different "levels of linguistic analysis".

2.1.1 Levels of Analysis

Nowadays, Natural Language Processing Systems are structured with different layers. The function of these layers resemble the different levels of language. But unlike a mere sequential model, the levels approach also takes into account that levels interact with each other dynamically. For humans, the understanding process has no strictly ordered sequence of steps, but uses information gained from other levels to make decisions on one particular level.

The following levels show different aspect of human understanding. All of them are used by humans to grasp the meaning of written or spoken language. A high-level NLP system should consider all aspects to ensure that no information is lost.

These first two levels only apply if we deal with spoken language. Often systems are specialized merely in *Speech Recognition* and leave the analysis of the recognized text to language analyzing systems. Anyway, these two levels are the starting point for humans to understand language.

Phonetics. This level deals with sounds and human voice. It is engaged in the different properties of speech sounds (*phones*) and their production, audition and perception.

Phonology. This level deals with the interpretation of speech sounds within and across words. It describes the way sounds function within a given language or across languages. The basic units on this level are minimal meaningful sounds (*phonemes*). And the task of this level is to distinguish different similar phonemes, and to take into account stress, accent, or intonation, which can influence the meaning of a word.

In a Speech Recognition system sound waves are analyzed and encoded into a digital signal for interpretation by various rules or by comparison to a particular language model.

Morphology. This level deals with the composition of complex words out of meaningful syllables. For example, the word "presupposition" consists of the *morphemes* "pre" (its prefix), "suppos" (the root), and "ition" (the suffix). For humans it is possible to grasp the meaning of such a composite word if they understand the single morphemes. NLP systems also make use of this fact, if it comes to determining the time an action took place, which is encoded in the suffix of the verb (usually "-ed" for past tense).

Lexicality. The interpretation of single words takes place at this level. For NLP systems this means assigning a part-of-speech (POS) tag to each individual word. If more than one POS tag is possible, the system has to assign the most likely one regarding the context of the word. On this level, it is also possible to replace single words that have only one meaning with a semantic representation. This may include a description of the words using abstract, simple notions. To accomplish this goal, some kind of lexicon is necessary that contains the possible part-of-speech tags encoded for a single word. Additionally it may also contain more complex information about a word like the semantic class, possible arguments it needs, limitations for these arguments, different senses of the words and in what context which sense is used.

Syntax. After the word level, this level deals with the structure of a sentence as a whole. The structure of a sentence contributes a lot to its conveyed meaning in nearly all languages.

For an NLP system the identification of sentence elements and their function within the sentence is performed by using a grammar and a parser. The result should be a tree with the words as leafs and dependency relations between these words encoded in the tree structure. Not all NLP applications need a full parse of each sentence, depending of the task they were designed for.

Semantics. This is the level where the sense of a word with ambiguous meaning is determined. To achieve this goal, the context can be useful, and of course the part-of-speech tag of a word. If there are still multiple senses possible after considering this information, statistical knowledge can be useful. The information of the pragmatics level might narrow the possible sense by giving information of the domain of a text. Although it is common to refer to semantic as meaning in an ordinary way of speaking, the semantics level contributes only one part to the meaning of a text.

Discourse. Now that we looked at words and sentences and analyzed the syntax and semantics for sentences, we advance to text units spanning more than one sentence. The information gained from looking at more than one sentence can be to identify different functions of sentences. For example in a dialog, the discourse partners utter different arguments, questions, and answers to the questions. This structure contributes to the meaning of the whole text and conveys information that can not be gained from looking at single sentences. Another task performed on the discourse level is the resolution of anaphora, that means replacing words like pronouns by representations of the actually referring entity.

Pragmatics. This level deals with additional meaning added to the text by meta information about it. Also the situation the text was created can be taken into account, as well as the purpose of the author. This task requires a lot of world knowledge and is hard to achieve for computational systems. Intentions, goals, or plans have to be understood, and therefore the ability of inference is necessary. The information from this level can on the other hand also be used to solve problems on the other levels.

We should spread out the active research field of NLP in more detail, therefore we take a look at the various applications that have been developed incorporating these levels.

2.1.2 Natural Language Processing Applications

Building an NLP system requires implementing several of these levels. Because an NLP system is very complex and requires lots of different individual subsystems, nearly all NLP systems are built in a modular fashion. Each of the subsystems or modules concentrates on one specific task. The re-usability of such modules in other systems is very high, and it is not necessary to start from scratch with every new system. Most NLP systems need modules from the lower levels, for example to identify sentences. Another advantage of modularity is the possibility to use different approaches for each module. This opens the way to use statistical approaches on well researched low-level tasks and symbolic ones for the higher levels which require world knowledge or sophisticated inference rules.

Each of these modules describes research areas that aim to improve the performance of the particular task. These subtasks within an NLP system include areas like:

- Anaphora Resolution,
- Named Entity Recognition,
- Coreference Detection,
- Part-of-Speech Tagging,
- Syntax/Semantic Parser,
- Noun Phrase Chunking,
- Word Sense Disambiguation,
- Speech Recognition,
- Language Generation,

An NLP application usually builds some higher level implementation upon existing components. Let's take a look at the applications using NLP techniques to solve problems.

Summarization: Summarizing large amounts of texts by condensing the contained important information. This is done using shallow approaches, statistical methods or rather high levels of language analysis to identify important information.

Machine Translation (MT): A typical NLP application that tries to translate text from one natural language to another.

Information Retrieval (IR):] The task of finding documents that contain certain information, or finding meta-data about documents, as well as searching in databases.

Information Extraction (IE): Similar to IR, Information Extraction is concerned with finding information in text, but IE goes one step further and tries to extract structured information found in various unstructured sources.

Question-Answering (QA): In contrast to Information Retrieval, which provides a list of potentially relevant documents in response to a user's query, question-answering deals with finding an answer in a corpus to a question posed in natural language.

Dialogue Systems: Using natural language to communicate with humans providing a better human-computer interaction.

The Fuzzy Believer Application can not be clearly classified as being in one of the categories, although Information Extraction would probably describe the function of the application best. But also some aspects of Question-Answering and Dialogue Systems are relevant.

2.2 Related Work

We split this chapter into two parts, belief extraction and belief processing. There is no clear boundary between these two categories because computational systems often do both. The single approaches presented are therefore classified according to their focal point.

2.2.1 Belief Extraction

This section deals with the extraction of information from text that could be interpreted as beliefs of any kind. Existing work on this field deals mostly with the extraction of opinions, for example customers opinions about a product. Others extract reported speech or general world knowledge, or try to find emotional statements.

2.2.1.1 Extracting Reported Speech

In book chapter "Conveying Attitude with Reported Speech" [Ber05], as well as in "Attributions" [BDGW04], a system is presented that extracts reported speech from newspaper articles together with the source and reporting verb for each reported speech occurrence. This information is passed through evidential analysis and presented in different *profiles*.

A newspaper article annotated with part-of-speech tags, noun phrases and verb groups is the starting point for the reported speech extraction component [Doa03].

A strict correctness measure yields a recall of 55% and a precision of 87% for the extraction process. A more lenient interpretation 63% recall, respectively 98% precision. These results were gained by analyzing 2404 sentences of Wall Street Journal articles, which contained 513 reported speech sentences. The successful extraction provides a *basic profile*, demerging source, reported verb, circumstantial information and reported speech.

To evolve *profiles* out of *basic profiles*, an intermediate step (*merged profiles*) are needed. To get them, the exploitation of coreference information becomes necessary. For this reason, a noun phrase coreferencer [WB03] is used to identify same sources of different statements. These statements are then merged into one *merged profile*.

A third and last step tries to build up evidential chains and includes a percolation algorithm [Ger00]. The merged statements are grouped according to the reporter who uttered the reported speech. This allows to model different degrees of confidence into a certain newspaper, a certain reporter, and a certain source. To encode the different confidences in the resulting profile, a dichotomy of held beliefs and potential beliefs is introduced. A statement in a reported speech construct first becomes a potential belief and later, presumed further evidence, an actually held belief.

2.2.1.2 Extracting Opinions

In [BYT+04] "Automatic Extraction of Opinion Propositions and their Holders" the authors present a system for identifying opinions in newspaper articles. They concentrate on what they call *propositional opinions*, that means opinions expressed in the propositional argument of certain verbs. Out of the two opinions:

(i) I believe in the system.

(ii) I believe you have to use the system to change it.

the second one states a propositional opinion: *you have to use the system to change it.*

To find these opinions, a semantic parser was used [PHW+03] to label the appropriate parts of sentences together with a lexical feature based on specific words related to opinions. The semantic parsing was done by training a statistical classifier, based on different hand-annotated training and test corpora. For the lexical feature, nouns, adjectives, adverbs and verbs were considered, found by comparing different types of texts and computing the relative frequency of words, as well as identifying co-occurrence with known opinion-related adjectives.

A one-tier and a two-tier architecture are used to implement the system. For the one-tier approach, the semantic parser is used to label parts of a text as either *opinion-proposition* or *null*. This baseline feature was extended with a feature set derived from the word lists gained before. Within the two-tier approach, the semantic parser is used to mark all *propositions* as the bottom tier, and different Naive Bayes classifiers on top to determine if the proposition expresses an opinion.

The achieved precision and recall for the best configuration for the one-tier architecture is 58% and 51%, for the two-tier architecture's best configuration it is 68% and 43%.

2.2.1.3 Identifying Opinions

Extracting opinions from customers reviews, blogs, or news sites has become a promising task. In [KH06] "Identifying and Analyzing Judgment Opinions" the authors present a new system for detecting and processing opinions. They mention other research efforts in this field, which make use of cue words to identify opinion holding statements as a main strategy. One problem with this approach is that words used to express an opinion in one domain can be used in another domain to merely state a fact. Opinions can have different shapes. They can mainly be described to be beliefs about the world on the one hand, or judgments about the world on the other. Because of the fact that even humans often disagree on whether a belief is an opinion or not, the authors concentrate on the second type of opinion: Judgment opinions. They consist of three parts: A *valence*, a *holder*, and a *topic* of the opinion. Their system works on word level opinion detection and for short emails.

The first step is to identify opinion bearing words and to classify their valence into *positive*, *negative*, or *neutral*. This is done by using manually annotated seed words and expanding this list using WordNet synonym classes and a formula to compute the closeness of the seed words to the synonyms.

The second step is to identify the entity who expresses the opinion explicitly or implicitly. Therefore, a Maximum Entropy model is used that ranks candidates and the highest ranked candidate gets chosen. This approach is better than using the ME model to classify the candidates. The manually annotated MPQA corpus[1] was used as training data to extract certain features like the syntactic path between a candidate and the opinion expression. The third step – identifying the topic – is not addressed in the paper.

The opinion detection and valence identification was used to extract sentiments from German emails. This can be done by either translating the opinion bearing words into German, or by translating the emails into English. The system was used with both strategies. Manually annotated test data was used to evaluate the system.

The results for the different parts show a high recall and varying precision for the identification of opinion bearing words, depending on the word class. For the extraction of the opinion holder, only an accuracy of about 50% was achieved, with a baseline algorithm achieving about 21%. For the email system, recall was between 40% and 80% for positive and negative emails using both translation methods. Precision was between 55% and 72%.

2.2.1.4 Mining Opinions

Analysing opinions is a relevant task to gather information about customer satisfaction. [GACOR05] presents in “Pulse: Mining Customer Opinions from Free Text” a system that not only uses a machine-learned sentiment classifier to detect positive or negative opinions but also incorporates a topic detection mechanism. The analysis is based on the sentence level.

The data the system was run on was a car review database with over 400.000 car customers reviews. 3000 sentences out of the database were manually annotated as *positive*, *negative*, or *other*. This data was split into 2500 sentences as training data for the classifier, and 500 as gold standard for evaluation purposes. Tests about inter-annotator agreement[2] revealed the difficulty even for humans to categorize the sentences.

The first step of the system is to extract a taxonomy by querying the car review database. The result are manufacturer and model names of cars. For each model,

[1] http://www.cs.pitt.edu/mpqa/databaserelease/
[2] between 70% and 80%

the sentences are extracted and different terms within the sentences chosen by assigning tf-idf values and using go- and stop-lists. The highest scored terms form a keyword and the sentences containing this word are clustered together. Beside that, the sentiment classifier assigns a positive, negative, or other value to each sentence. The also implemented visualization component shows for each car model boxes labeled with the keywords. The size of the box corresponds to the number of sentences within the cluster, and the color of the box the average detected sentiment.

Because of a predominance of positive reviews in the corpus (over 60%), recall and precision for the negative class is more interesting: Recall is between 10 and 24% and precision is between 55% and 85%. If the amount of data is big enough, low recall is acceptable.

2.2.1.5 Extracting World Knowledge

"Extracting and Evaluating General World Knowledge from the Brown Corpus" [ST03] presents a technique to extract world knowledge out of miscellaneous texts. The approach is to apply interpretive rules to clausal patterns and patterns of modification, and concurrently abstract general probabilistic propositions from the resulting formulas.

As input for the system, Penn Treebank like annotations are used with the original text to produce a logical form of the found patterns. The whole system consists of six components:

(i) A Treebank preprocessor

(ii) A pattern matcher

(iii) A semantic pattern extractor

(iv) Abstraction routines

(v) A Propositional patterns deriver

(vi) Filter heuristics

The main task is to evaluate the output *general world knowledge*. This was done by judges using an evaluation software, classifying the generated propositions into: reasonable, obscure, vacuous, false, incomplete, or hard. The different judges classified 49–64% of the proposition as reasonable. The lack of a good enough parser producing correct parse trees needed for the pattern extraction is the main problem.

2.2.1.6 Finding Emotional Verbs

Verbal expressions are rich bearers of emotions, feelings, and attitudes towards certain facts. In "A Semantic Lexicon for Emotions and Feelings" [Mat04] a lexicon is presented to categorize words for emotions or psychological states. Words belong to semantic classes that are linked with each other by meaning, intensity, and antonym relations. A graph structure is used to connect different classes.

Verbs within one class are assigned different properties, like the different modes of employment, different arguments, or the form of an expression containing the verb. This hand-generated lexicon offers the possibility to analyse emotional verbs. It can be used to find synonyms, antonyms, paraphrases, or intensity variations of one word.

2.2.2 Belief Processing

In this section we present systems and approaches dealing with belief processing. This comprises generating new knowledge out of existing information or comparing different pieces of information with each other. An important task within belief processing is the capturing of the meaning of a sentence, especially the meaning of the main verb. The verb is a main factor to identify the relation of two sentences to each other. This is important for finding similar sentences or entailment.

2.2.2.1 Classifying Verbs

A system trying to combine lexical classification and formal representation is described in “Formal Semantics Of Verbs For Knowledge Inference” [Boy06]. It is based on Universal Semantic Code [Mar01] and provides a functional classification of verbs. The main point of the USC model is to consider verbs as actions and the consequences of actions can then be inferred.

The basic assumption of the model is that every statement contains at least implicitly a verb, and every verb represents an action. Verbs, and therefore also actions, are considered the main elements for world description, suggesting a strictly causal world where everything was caused by something. Verbs are divided into classes with same meanings, like “fill”, “charge”, or “inflate”. Each of these classes represent a certain action.

Together with an action, a formula represents the possible relations following with the action, for example, the formula $((X \to Y) \to Z) \to (Z \to (Y \to !W))$ can be interpreted as “A worker fills oil into the tanker by the loading arm”. USC axioms now define the transformation between different formulas. This can be represented by a directed graph connecting the formulas. The arcs are the USC axioms and following arcs through the graph means infer one formula from the other.

The system can be used to complete chains of action by detecting missing parts in the logical order of events. It contains 96 different *action classes* with 5200 entities.

2.2.2.2 Measuring Relatedness

Semantic relatedness between pair of words is hard to define. In “Measuring Semantic Relatedness Using People and WordNet” [Kle06], a data-set is proposed to test relatedness between words together with a new metric to measure relatedness.

A human produced data-set is necessary to evaluate different measures of relatedness. This data-set should be more than simple synonyms and was therefore designed to show different degrees of relatedness by marking different concepts that help the reader to understand other concepts. These concepts are considered to be related with each other. The annotator makes use of common knowledge to identify these kinds of relations between concepts. The degree of relatedness between a pairs of words is determined by the number of annotators who identified the relation.

For their measure, they propose the use of glosses in addition to the WordNet taxonomy. This should enable the metric to detect relatedness across part-of-speech boundaries, for example between “kill” and “death”.

The evaluation shows that this measure works better than other state-of-the-art similarity measures for the human generated data-set. The data-set itself does not contain information about a special relation but about relatedness in a more abstract way.

2.2.2.3 Learning Entailment

The Pascal RTE Challenge [DGM05], [BHDD+06] has led to the development of a number of new systems dealing with inference or entailment (see also Section 6.2.1). But unlike the simple approach looking for semantic overlap, [MGdM+06] "Learning to recognize features of valid textual entailment" presents a promising approach of graph alignment. To overcome some limitations related with this approach, they present an improved architecture, dealing with these limitations.

The simplest approach, looking only for semantic overlap, using n-grams, bags of words, or TF-IDF scores, are not well suited for the inference task. Entailment is not, unlike semantic overlap, a symmetric relation, and the overlap measure does not address the syntactic or semantic structure of the sentences. The graph alignment approach tries to find matching graphs between the dependency parse trees of the two texts to be compared. The problem of partial graph alignment is NP-complete, leading to attempts to approximate the search process. Another approach translates the sentences into quasi-logical form, and tries to match the resulting nodes, with or without additional background knowledge. One of the graph matching approach's weaknesses is that it assumes an upward monotonicity, not considering context information, and only looking at matching parts of the text and the hypothesis. The assumption of locality can also bear problems, as well as the inherent confounding of alignment and entailment determination.

To overcome these three problems of the graph alignment approach, the authors [MGdM+06] propose a two-stage architecture, separating the alignment process from the entailment determination. After finding a good alignment, a classifier using some global features from the alignment graph can decide, if there is entailment or not. The weights for the features can be determined using training data and machine learning techniques. The classifier uses 28 different features for entailment detection or exclusion. Especially the shortcomings of the graph alignment approach mentioned above can get solved.

The evaluation for the system on the Pascal RTE Challenge data shows that the confidence weighted score (CWS) on the development data is 59.1%, 65.3%, and 74.4% for the system using only alignment, using hand-tuned feature weights, and learnt feature weights. For the test set, the numbers are between 60% and 65%, which is not significantly higher than for other systems (~64%).

2.2.2.4 Recognizing False Entailment

Textual entailment is a hard task for NLP systems. In [SVM06] "Effectively Using Syntax for Recognizing False Entailment" the authors present a system that uses heuristics to determine entailment. The focus lies on heuristics to identify false entailment. While the inter-annotator agreement is usually very high (95,1% [BM05]), automated systems perform rather poorly (around 50% [DGM05]). Other systems like this one use logical form representations, but unlike this one, they start with the premise that the entailment is false and look for hints of true entailment.

Here, the premise is that the sentence in question entails the hypothesis. The system tries to detect false entailment with the use of different heuristics that find: unaligned entities, negation mismatches, modal auxiliary verb mismatches, antonym matches, argument movements, superlative mismatches, and conditional mismatches.

These heuristics make use of a word alignment structure built out of a computed *logical form*. This is done using the Microsoft NLPWin system. If none of the heuristics above discover false entailment, the system computes a similarity measure for the aligned words using MindNet[3] and for phrasal similarity the web.

The result for the system on the PASCAL RTE challenge [DGM05] data was an accuracy of 62,5%.

2.2.2.5 Classifying Agreement/Disagreement

A semi-supervised learning method for a contrast classifier to distinguish between agreement and disagreement is shown in “Agreement/Disagreement Classification: Exploiting Unlabeled Data using Contrast Classifiers” [HLO06]. Other semi-supervised learning algorithms have problems with unevenly distributed data, which the proposed method overcomes. It implicitly models different distributions of classes in the labeled and unlabeled data sets.

The classifier is used to identify a simple speech act, namely the difference between agreement and disagreement. For each class the classifier is trained on a balanced data sets. To compute the final output, a committee of classifiers is used. The contrast classifier works on each class independently and therefore models implicitly the distribution for this class.

The contrast classifier approach performs better than co-training and self-training in detecting the infrequent classes.

2.2.2.6 Computing Polarity

In the paper “Computing relative polarity for textual inference” [NCK06] the authors present a strategy to detect the attitude a writer of a sentence has towards the truth or falsehood of the complement clause of this sentence. This is based on the syntactic type and on the meaning of the embedding predicate.

A writer who writes the sentence “Ed forgot to close the door”, is committed to also believe the sentence “Ed did not close the door”. In this case, the embedded predicate “forget to” entails not doing it. This is not always true because it relies on the type of the complement clause. These implicative verbs have been analyzed as well as the effects of recursive use of embedded predicates, like in the sentence “Ed didn’t manage to remember to open the door.”.

Out of 1250 relevant verbs, 400 were classified on what kind of implicative verb they are and what kind of entailment can be inferred. The system they implemented to recognize entailment based on their analysis, uses the XLE parser [MK96] for syntactical parsing and “skolemization” to build up a semantic representation of the sentences. These structures are summarized to more uniform representations. In addition, an algorithm is used to deal with the recursion of embedded predicates.

Unfortunately, the paper does not contain information about the performance of the system or the algorithms, maybe because it deals with a very particular feature of language.

2.2.2.7 Modeling beliefs

“An argument-based framework to model an agent’s beliefs in a dynamic environment” [CCS05] uses logic programming to represent the knowledge and of an agent in a multi-agent system (MAS) environment. They define *Observation based Defeasible Logic Programming* to represent and to reason about the beliefs of an agent. Especially the processing of new input data that might change the agents current

[3] http://research.microsoft.com/mnex

beliefs is considered. To speed up the reasoning process, a *dialectical database* is introduced that stores pre-compiled knowledge.

Defeasible Logic Programming (DeLP) provides a language for knowledge representation and reasoning using defeasible argumentation to decide between contradictory conclusions through a dialectical analysis. Within a MAS scenario, the dynamic aspect of new information has to be considered. Therefore an update process adds new information and solves possible inconsistencies within the agent's belief base.

The application of pre-compiled knowledge as done in Truth Maintenance Systems, is also possible to speed up the knowledge processing in a MAS setting. For a new query, not all existing arguments have to be recomputed allowing the agent to work in a real-time environment.

There is no evaluation of the system, but pre-compiled knowledge can speed up the reasoning process.

2.2.2.8 Revising Beliefs

Belief revision and argumentation theories are two research areas connected with each other. In "Revising Beliefs Through Arguments: Bridging the Gap between Argumentation and Belief Revision in MAS" [PC05] the authors try to show the common ground of both in a multi-agent-system-based framework. While belief revision deals with the changing of an agent's own mind, argumentation deals with changing the mind of other agents.

The authors claim that the existing AGM ([Gä88]) model of belief revision is not suitable for argumentation. It allows only three types of change within an agent's belief base (expansion, contraction, and revision), its belief base are sets of propositions, and postulates ensure the rationality within the process. The problem why this is not usable for argumentation is the necessary ordering introduced on the initial belief base. This does not allow a logic argumentation on why a certain belief is hold by the agent. It is on the contrary only held because it fulfills the postulates for a consistent belief base.

The paper now suggests to divide the information an agent has into *data* and *beliefs*. This allows to make a distinction between gathered data, which may become belief later, and actually held belief. Depending on factors like *relevance*, *credibility*, *importance*, and *likability*, beliefs get formed out of the data and get assigned a *strength*. Information about the source and their reliability for example are important variables influencing the credibility of the proposition, also the other way around: the reliability of a source rises if it utters believed propositions.

The argumentation part shows that the proposed model can be applied to the most influential model of Toulmin [Tou03].

2.3 Discussion

As we saw at the beginning of the chapter, an NLP system usually consists of a series of components. For effectively developing new NLP applications, it is therefore reasonable to use existing components build for a specific task and only develop high level specialized components. Our system, as well as most other NLP systems described here, follow this approach, and incorporates external components within its structure. The crucial part of our implementation that distinguish our work from the different NLP applications dealing with summarization, information extraction, or for example machine translation, are the high-level components especially developed to build a Fuzzy Believer system.

The task for belief systems, in contrast to summarization, is rather vague. There is no clearly defined goal of the research community, as what a belief system has to look like, what its domain of application is, or how it should work in general. That is one of the reasons, why nobody, as far as we know, has designed *and* implemented a comparable system. There are a couple of theoretic approaches on how to deal with beliefs computationally like [WF94], [KS00], or [LF06]. But because of the difficulty for computational systems to grasp meaning of text, implementations are rather scarce. The areas of *knowledge representation* and *computational logic* have developed more or less complex theories and systems that cover their specific field, for example [CC04] for belief logic and neural networks, or [LG91] for a knowledge representation language (CycL). The challenge is to build a system that incorporates all these aspects and is of practical use.

In the section above, a variety of systems were presented dealing with similar aspects, but either with the focus on belief extraction or belief processing. Our system was designed to solve both tasks, making the processing component dependent on the quality of the extraction part. Additionally, the systems have been developed to achieve very different goals. Belief processing for the general domain of newspaper articles is a goal not many systems try to tackle, and often other systems limit their domain of application to special areas. We are going to show that it is possible to develop a system that can deal with the general domain of newspaper articles.

Chapter 3

Theory and Foundations

This chapter is divided into three parts, in which we give an overview of the theoretic background of this work. The first part is dedicated to *reported speech*. As we have seen in the example newspaper article in the introduction (Figure 1.1), opinions about a topic are often presented by a reporter using reported speech. The idea for our Fuzzy Believer system is to exploit these reported speech structures to generate beliefs. To extract reported speech automatically, it is important to know its structure, its way of occurrence, and what information can be encoded in it. A deeper analysis of what exactly is reported speech, how it is used, and what kinds of appearances exist, is given below. Further reflexions on related considerations about reported speech (*evidential analysis* and *speech acts*) show the usage and the information richness of reported speech.

The subsequent part deals with background information of the consecutive computing step of the Fuzzy Believer system to interpret the reported speech information as a beliefs. We will see in this section what is necessary to make a *belief* out of statements found in reported speech. "Belief" is not a well-defined notion but a rather general concept used to express a certain state-of-mind or attitude towards a proposition. We try to narrow down the meaning of "belief" to an acceptable definition that allows us to employ this notion for the purpose of natural language processing. Our focus will lie on the representation of belief structures and how these structures can be generated and used by computational systems. So before the Fuzzy Believer system starts processing beliefs, a representation of the reported speech information has to be generated that the system can handle.

The last part of this chapter gives an introduction to *fuzzy set theory*. This is the core concept used in our system to process the generated belief structures. Fuzzy set theory notions and definitions are presented, providing a basis for our approach. To employ fuzzy processing to the extracted beliefs, the belief structures have to be transformed to a fuzzy belief representation. This representation enables us to use standard fuzzy operators presented at the end of this part to process our beliefs. Defuzzification at the end of the process yields the result of the system comprising accepted and rejected beliefs.

We will show in this chapter the foundations for the different computing steps performed by our Fuzzy Believer system. The evolution from the input to the output of our system can be summarized as the transformation and processing from newspaper articles to beliefs:

Article $\rightarrow$ Reported Speech $\rightarrow$ Statements $\rightarrow$ Beliefs $\rightarrow$ Fuzzy Beliefs $\rightarrow$ Believed Statements.

3.1 Reported Speech

In this section, we will give an introduction to reported speech. Unlike the use of the linguistic term of reported speech[1], which denotes only indirect speech, in contrast to direct, or quoted speech, we will refer to reported speech as both of them. Both types contain information reported by a third person – communicating between source and recipient. For the purpose of ascribing a statement to a source, it is not necessary to distinguish between direct and indirect speech. For a more detailed analysis later on, the difference, however, does matter.

3.1.1 Overview

Most of our knowledge originates not from our own experience, but from facts someone told us. To add weight to one's statement and increase the credibility of facts, it is often necessary to name the source of the original statement. This is also an option if the speaker wants to dissociate himself from the statement, making clear that he disapproves the statement. Reported speech is the vehicle to express different statements originating from someone else, while at the same time one's own attitude towards this statement can be clearly stated.

The most extensive use of reported speech can be found within newspaper articles. Newspaper articles should contain authentic facts, backed up by the original source. A journalist is not supposed to make something up by himself or to report on an event only from his subjective point of view. Therefore, he has to rely on statements from others and this has to be pointed out in the article.

In many articles, reported speech sentences make up to 90 percent [Ber92] of the whole article, showing the importance of this mode for reproducing language of others. Thus, newspaper articles are well suited to explore reported speech.

3.1.2 Structure

Reported speech usually consists of the *reporting clause* and the *reported clause*, [Qui85]. The reporting clause contains information about the speaker or source of an utterance, as well as the way the utterance has been done (*Fred said*; *The report stated*). It can also contain information to whom the utterance is addressed or in what circumstances the utterance is done:

$$\underbrace{\overbrace{\text{Last October}}^{\text{circumstances}}\text{, }\overbrace{\text{his brother Hubert}}^{\text{source}}\ \overbrace{\text{told}}^{\text{reporting verb}}\ \overbrace{\text{the bankruptcy court}}^{\text{addressee}}}_{\text{reporting clause}}\text{ that }\underbrace{\text{Paul was very ill}}_{\text{reported clause}}.\text{[2]} \tag{3.1}$$

The reported clause can consist of direct speech or indirect speech. The same example as in (3.1) now with direct speech as reported clause:

$$\text{Last October, his brother Hubert said:“Paul is very ill”} \tag{3.2}$$

There is also a form of *free direct and indirect speech*. It is used for example to express a stream of consciousness in fictional writing. The reporting clause is omitted in that kind of reported speech. In newspaper articles, the free forms do not occur, therefore in the following discussion we will concentrate on direct and indirect speech.

[1] see [Qui85, page 1021] for a definition

[2] from Wall Street Journal 03.03.88

add	comment	object	say
admit	conclude	observe	shout
announce	confess	order	state
answer	cry	promise	tell
argue	declare	protest	think
assert	exclaim	recall	urge
ask	explain	remark	warn
bet	insist	repeat	whisper
boast	maintain	reply	wonder
claim	note	report	write

Table 3.1: Reporting verbs of speaking and thinking that are frequently used within direct speech (from [Qui85, page 1024])

Direct Speech. Quotation marks usually indicate direct speech. In the domain of newspaper articles, quotation marks are obligatory[3]:

In a statement yesterday, Towle President Paul Dunphy said,
"We look to a closing in the very near future" (3.3)

Instead of the colon it is also possible to use a comma to separate the reporting and the reported clause (compare (3.2) with (3.3)). The position of the reporting clause can vary. Apart from its occurrence at the beginning of a sentence, as in the example above (3.3), it might also occur in medial position, as in (3.4), or at the end of the sentence (see (3.5)).

$$\text{"Our deepest concern,"} \left\{ \begin{array}{l} \text{the President said,} \\ \text{said the President,} \end{array} \right\} \text{"is the rising unemployment rate."} \quad (3.4)$$

When the reporting clause is located within or after the reported clause, subject and verb may exchange their positions, if the reporting verb is in simple present or simple past tense. This construction is typically applied if the position of the reporting clause is in the middle, the subject is not a pronoun, and the reporting verb is *said*, as in example (3.4).

"We think this is the bottom year," a Nissan official said.[4] (3.5)

Direct speech can span over more then one sentence. In that case, the reporting clause is usually found within the first sentence.

As a grammatical relation, the direct speech can function as a subordinate clause, for example in (3.3). But it can also be a subject complement, an apposition to a direct object, or an adverbial construct.

A special case is the mixture of direct and indirect speech. The direct speech is only a part of the reported clause and is integrated as a partial direct speech within the reported clause:

In a televised address, the president concluded that the initiative "was a mistake."[5] (3.6)

Table 3.1 lists verbs that are frequently used within direct speech. Apart from these verbs, verbs indicating the manner of speaking like *mumble*, *mutter*, or *sob* can also be used within the reporting clause of direct speech.

[3] from Wall Street Journal 09.14.87
[4] from Wall Street Journal 12.17.86
[5] from Wall Street Journal 03.05.87

acknowledge	boast	declare	mention	report
add	certify	deny	object	retort
admit	claim	disclose	predict	say
affirm	comment	exclaim	proclaim	state
agree	complain	explain	promise	submit
allege	concede	forecast	pronounce	suggest
announce	confess	foretell	prophesy	swear
argue	confide	guarantee	protest	testify
assert	confirm	hint	remark	vow
bet	contend	insist	repeat	warn
	convey	maintain	reply	write

Table 3.2: Factual verbs, *public* type, of speech acts concerned with statements that are frequently used within indirect speech (from [Qui85])

accept	doubt	imagine	realize
anticipate	dream	imply	reason
ascertain	ensure	indicate	recall
assume	establish	infer	reckon
believe	estimate	insure	recognize
calculate	expect	judge	reflect
check	fancy	know	remember
conclude	fear	learn	reveal
conjecture	feel	mean	see
consider	find	note	sense
decide	foresee	notice	show
deduce	forget	observe	signify
deem	gather	perceive	suppose
demonstrate	guess	presume	suspect
determine	hear	presuppose	think
discern	hold	pretend	understand
discover	hope	prove	

Table 3.3: Factual verbs, *private* type, of speech acts concerned with statements that are frequently used within indirect speech (from [Qui85])

Indirect Speech. Especially in newspaper articles, indirect speech is used to report statements:

As a result, the company said that it will restate its 1986 earnings.[6] (3.7)

As in (3.7) indirect speech can be a direct object, a extra-posed subject, or a subject complement. Indirect speech, as well as direct speech, makes use of the reporting verbs shown in Table 3.1. Further, usually factual verbs are used to introduce indirect speech. In Table 3.2 the most common factual verbs, which express speech acts that are accessible for the *public*, can be found. In Table 3.3 the *private* type factual verbs are listed. They describe speech acts that are only accessible and verifiable by the one who makes the utterance.

Indirect speech can to a certain extend summarize or paraphrase the original utterance, hopefully without changing the meaning. By using indirect speech, the deictic features of the language have to be altered. They refer to certain time and location of the speaker, and are usually not applicable by the reporter. The original

[6] from Wall Street Journal 02.05.87

utterance may also contain references to other persons that are not clear without context. Things to be changed within indirect speech therefore include [Qui85]:

(i) Tensed forms of the verb.

(ii) Other time references, e.g.: *yesterday*, *now*, *last week*, *next Monday*.

(iii) Place references, e.g.: *here*, *there*.

(iv) Personal pronouns.

(v) The demonstratives *this* and *these*.

The tense usually has to be back-shifted, by replacing present tense with past, present perfect with past perfect, and so on. There are exceptions of this rule, for example, if the statement in the original utterance is a universal truth, the back-shift of the tense is optional:

Teacher: "The world is round!"

(i) The teacher said that the world is round.

(ii) The teacher said that the world was round.

Also, the personal pronouns have to be adapted:

Paul: "I owe you 10 Dollar."
Paul said that he owed me 10 Dollar.

3.1.3 Speech Acts

Austin [Aus62] and Searle [Sea69] have introduced *Speech Acts* to the linguistic community. Speech Acts lay weight on the fact that everybody saying something also is doing something implicitly, like threatening s.o. or cheering s.o. up, etc. We do not want to go into details about Speech Act Theory but point out a special case concerning reported speech.

Imagine the following situation: The President says in front of the Parliament:

"I will not sign this bill." (3.8)

With saying that, he implicitly denied to sign the bill. A newspaper reporter might then write an article containing the sentence:

The President denied to sign the bill. (3.9)

Grammatically, the reported clause would be "to sign the bill" which is not a complete sentence. To get a complete sentence, the text has to be rephrased. Without knowing the actual utterance of the President, a couple of variants are possible:

The President said that he denied to sign the bill. (3.10)

The President said: I didn't sign the bill. (3.11)

The President said: "No." (3.12)

The President shook his head. (3.13)

The meaning of these sentences are not the same as the original one. It can not be inferred from the original reported speech structure what has actually been said, thus we can not reconstruct out of a *reported speech act* an original utterance, even if there was one.

3.1.4 Evidential Analysis

How can information gained from reported speech be exploited? By choosing a certain reporting verb, the reporter tries to encode the original speech act. Further more, the chosen reporting verb can contain information on how reliable the source is. The reporter can use the mechanism of reported speech to not only reproduce the content of the utterance, but to reproduce and clarify the whole speech act.

From the reader's point of view, reading a newspaper article is a multilevel process. The reader has to:

1. Understand the content of what is expressed in the article (the reported clause).
2. Evaluate the additional information given by the reporter to reconstruct not only the original utterance but the original speech act (the reporting clause).
3. interpret the article as presented by the reporter.
4. reconstruct the original situation.
5. interpret the assumed original situation.

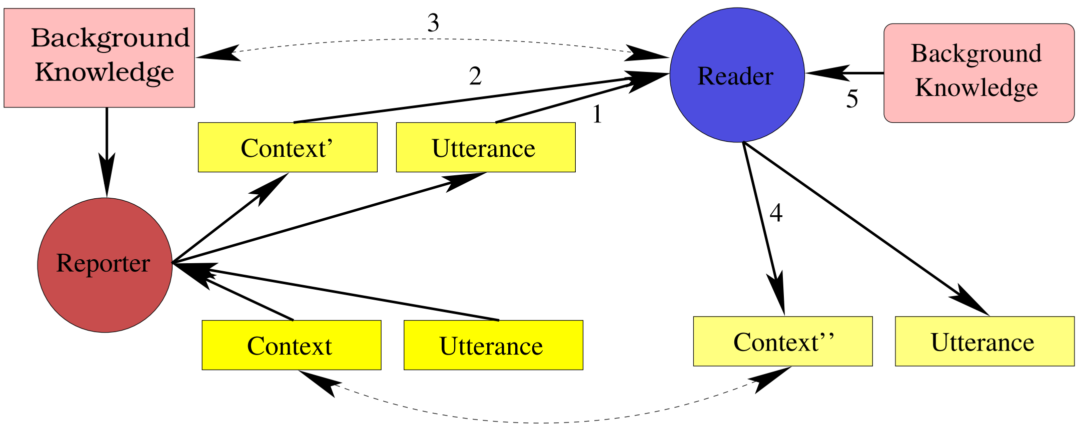


Figure 3.1: Reported speech

Bergler presents in [Ber92] a detailed analysis of the different aspects and semantic consequences of reported speech. The core statement is that reported speech contains information about the evidence of the reported clause for the reporter. In the case of newspaper articles, the reported clause usually contains *primary information*, as opposed to solely *circumstantial information*. The latter one is used to support previously made statements, whereas primary information introduces new facts, positions, or actions. The reporter wants to present this primary information to the reader, but also has to encode the reliability of the source within the reported speech. As another level of indirection, even the source may not be completely convinced of what he/she is uttering. All these different attitudes have to be evaluated by the reporter, who has maybe even more background information or statements by other sources that has to be taken into account. The different levels of trust and reliability lead the reporter to the final phrasing of the reported speech. To express his confidence in a statement, the reporter can first decide which statement to mention, he can choose a reporting verb that suits his purpose, and finally find a description of the source that encodes its reliability. This observation allows the summarization of the evidential analysis as stated in Figure 3.2.

Evidential Analysis

$$\text{SAY(Source}, \phi) \text{ is true} \Longrightarrow \phi[OC, CC, TC] \text{ is true.}$$

with OC = original context,
CC = current context,
TC = temporal context.

Figure 3.2: Inferential impact of the evidential analysis of reported speech

To stress the difference between evidential and intensional analysis, we look at the philosophical impact of both: For the intensional analysis, the truth value of the utterance can not be decided at all, for the evidential analysis, on the other hand, the truth value of the reported clause can be inferred from the circumstantial information and its reliability. This allows commonsense reasoning over the facts mentioned in the reported speech, with the need to keep the information contained in the reporting clause, because reliability, confidence into a source, or fundamental beliefs might change. In this case, a re-evaluation of the reported speech becomes inevitable.

The manifold information that can be found encoded in reported speech enables the Fuzzy Believer system to choose which information to focus on. The reported clause containing the actual statement is the most important one. The analysis of it and the processing to gain *beliefs* is the main task of our system. The following section is concerned with this very notion of "belief".

3.2 Belief

First of all we have to define what we mean by *belief*. In the Webster's Revised Unabridged Dictionary ([Por13]) belief is defined as:

> Assent to a proposition or affirmation, or the acceptance of a fact, opinion, or assertion as real or true, without immediate personal knowledge; reliance upon word or testimony; partial or full assurance without positive knowledge or absolute certainty; persuasion; conviction; confidence; as, belief of a witness; the belief of our senses.

A belief can be true, false, or neither of them. We can believe things that can not be decided to be true or false, we can believe things to be true although they are not and vice versa. Even if we know what we belief, we do not know about belief itself.

To model the concept of believing for a computer we have to dive into the field of artificial intelligence. A machine that is capable of holding certain beliefs is perceived significantly differently from a computer only doing "computations". The artificial intelligence research community has used this difference to define their subject of study. For this purpose, Turing ([Tur50]) proposed a test involving a person having a conversation with either another person or a computer. If the computer has artificial intelligence, the person should not be able to tell if he is talking with a computer or another person. If a machine can behave like a human by talking like one, does it matter, whether the inner processes of this machine is comparable to human ones? And are we allowed to use words like “think”, “believe”, or “feel” to describe these inner processes of an artificially intelligent machine? But these questions are already biased towards a special philosophical school (see for example Wittgenstein [Wit77]) which emphasizes on the behavior of a being and not on the “inner” structures. And the topic was a playground for different philosophical positions during the last centuries leading to a variety of different positions within the philosophical community.

3.2.1 Philosophical Considerations

There are philosophers who reject the possibility of machines having beliefs, or other human-like, intelligent attributes, totally. For them, *having a belief* means having a mental state representing this belief. And machines or computers are not able to have mental states ([Sea80]) at all. But to answer the question whether computers can have beliefs or not, it is crucial to define *belief* exactly. And there the differences between artificial intelligence research and philosophy, and even within the field of philosophy, are encountered. The artificial intelligence community does not reject the possibility of computers having beliefs. In fact, every single piece of data stored within a computer can be interpreted as a belief held by this computer. This is a rather rudimentary approach and does not involve any processing of beliefs or any contextual world knowledge. But this view is far away from what humans call having beliefs. So as an additional requirement, computers should be able to ascribe belief to other entities, enabling them to understand that other computers or persons can have different beliefs.

3.2.2 Representation

In this section we develop a strategy to structuring and storing beliefs. We do not know enough about human mechanisms to store and structure beliefs and this is also the reason for different approaches within the field of AI.

As a first idea, we could put all beliefs, in the shape of propositions, together on a big heap. And whenever we are looking for something we search or sort the whole heap. To process beliefs we rely on a sophisticated logic to do all the work. For an artificial believer this means storing sentences that are believed into a big data base.

In 1975, Minsky presented his *frame* theory ([Min75]). He claimed that belief, or more general, knowledge should be sorted by topics. All beliefs related to a topic were sorted into a *frame*. It was also Minsky who introduced in [Min68] the idea that humans have to have models of each other to do reasoning.

Ballim and Wilks ([BW91]) used these ideas to model an artificial believer. They identify three important ideas[7] for a model of a believer:

1. It must contain substructures or environments consisting of *relevant* items for a topic. Only thus can the *heap* view of reasoning be avoided.

2. Those environments can correspond to other believers as well as to inanimate entities(e.g., “cars”, “gas prices”, “architects”, “Albuquerque”).

3. The model (human or artificial) must be able to *nest* environments so as to model the way that the base believer can model the beliefs of other believers down to any level a situation requires.

[7] see [BW91, page 28].

User: Frank is coming tomorrow, I think.

System: Perhaps I should leave.

User: Why?

System: Coming from you that is a warning.

User: Does Frank dislike you?

System: I don't know, but you think he does and that is what is important.

Figure 3.3: A sample dialog between the system and a user

These ideas guaranty a certain structure for the different beliefs. Every belief about one special entity is associated with this entity's environment. Those environments contain not only the system's beliefs about this entity, but also what the system believes the entity believes. To illustrate this fact, Ballim and Wilks propose a visual design to emphasize the nested environments.

If we look at the sample dialog[8] in Figure 3.3, the system must hold certain beliefs to be able to react the way it does in the dialog. To model this particular set of beliefs, the visual representation, showing the belief environments of the system, might look like the one in Figure 3.4. The diagram expresses the structure of what the system believes the user believes Frank believes about the system.

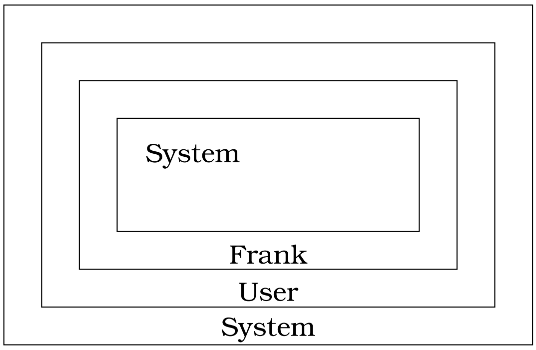


Figure 3.4: Sample belief environment structure of the system

The different placements of the labels express different functions: Labels in the top left corner denote a topic, labels on the bottom of a box represent an entity that has beliefs.

A system called ViewGen ([BW91] Chapter 4 and 5) has been developed to exemplify the work with different environments and beliefs about beliefs, etc. Within this system, the single box of an entity is called a viewpoint of this entity. Beside the different beliefs what other entities believe, the actual beliefs about an entity is also represented in the diagrams. In Figure 3.5 it is shown that the system not only has beliefs about beliefs of John, but also beliefs about John himself (his height, his gender).

As a basic algorithm [WB83] to process and work with the belief structure, ViewGen uses a default rule for the ascription of beliefs. A viewpoint gets generated by

[8] see [BW91, page 32].

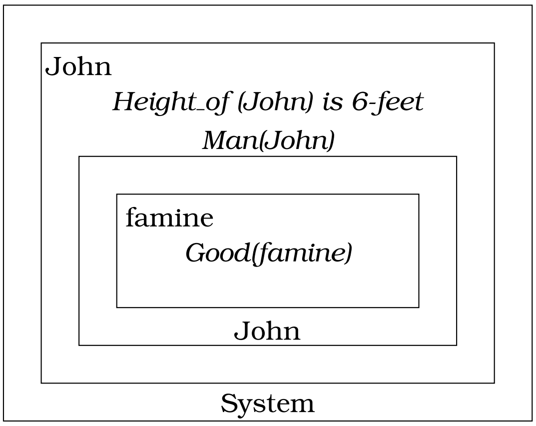


Figure 3.5: Sample believe structure of John and of the system about John

assuming, "... that the other person's view is the same as one's own, except where there is explicit evidence to the contrary.[9]"

For our system, especially the way beliefs get displayed is interesting. And, in contrast to ViewGen, our system's focus lies on the actual generation of the belief contents, not on the ascription of beliefs to different acting entities. To generate these belief contents out of reported speech we will employ fuzzy set theory, elaborated on in the following section.

3.3 Fuzzy Systems

Fuzzy set theory constitutes the main foundation of the processing in our system. Therefore, we introduce in this section concepts and notions of fuzzy systems necessary to understand our approach. The book "Architektur von Fuzzy Informationssystemen" [Wit02a] forms the basis of this part, and can also be used as a reference for more detailed background information.

Imperfect Information. With developing language, humans also have developed certain concepts to describe the world in which they live. Many of these concepts do not have a clear definition. As an example, we take a look at the concept of "heap". Already the ancient Greek philosophers had problems with this notion. It is not clear what constitutes a "heap", how many grains of sand are needed to get a heap of sand. This led to a class of paradoxes [Hyd05].

To solve these kinds of problems related with certain concepts, the notion of vagueness was introduced. It offers the possibility not to decide between wrong or right that means between "x is a heap" or "x is not a heap", but to assign a degree of "heapness" to x.

To deal with such vague concepts mathematically, Zadeh [Zad65] has introduced fuzzy set theory.

[9] see [BW91, page 43].

3.3.1 Fuzzy Sets

Definition 3.3.1 (Fuzzy Set) *A fuzzy set is characterized by a membership function μ_A, which maps the members of the universe Ω into the unit interval [0,1]:*

$$\mu_A : \Omega \rightarrow [0,1].$$

The set of all fuzzy sets over Ω is called $F(\Omega)$

To compute the degree of membership for element ω to set Ω it is usually not useful to use the whole unit interval. For a user, it is much easier to grasp the semantic meaning of a more coarse classification. For example, a scale with five different membership degrees is often well suited:

$$\text{“}certain'' > \text{“}likely'' > \text{“}possible'' > \text{“}unlikely'' > \text{“}impossible'' \tag{3.14}$$

Definition 3.3.2 (Normalized Fuzzy Set) *A fuzzy set is called normalized, iff:*

$$\exists \omega \in \Omega : \mu_A(\omega) = 1.$$

Definition 3.3.3 (Possibility Distribution) *Given a variable ω out of Ω, a function*

$$\pi_A : \Omega \rightarrow [0,1],$$

that meets the normalization property, is called a likelihood distribution over Ω.

Likelihood distributions and fuzzy sets are very similar. The difference lies in the interpretation of both: A likelihood distribution is a *flexible boundary* for variable x. $\pi_x(\omega)$ is the degree of likelihood for x being equal to ω. Therefore it holds that, if $\pi_x(\omega) = 0$, $x = \omega$ is impossible, and if $\pi_x(\omega) = 1$, $x = \omega$ is possible.

To store or process fuzzy sets, we need a suitable form of representation. One possible representation is the characterizing function of a fuzzy set. This can be a linear function or another parameterized function. The resulting representation is called *vertical.*

Another representation is called horizontal representation. It is generated by so-called α-cuts.

Definition 3.3.4 (α-cuts) *Given a variable $\mu \in F(\Omega)$ and $\alpha \in [0,1)$. The set*

$$[\mu]_\alpha = \omega \in \Omega | \mu(\omega) \geq \alpha$$

is called the α-cut of μ. The strict α-cut is defined by the set

$$[\mu]_\alpha = \omega \in \Omega | \mu(\omega) > \alpha.$$

Two special α-cuts are the core and the basis of a fuzzy set.

Definition 3.3.5 (Core and Basis) *The core μ_1 of a fuzzy set μ is the α-cut $[\mu]_\alpha$ with $\alpha = 1$. The basis $\mu_{>0}$ of a fuzzy set is the strict α-cut $\mu_{>\alpha}$ with $\alpha = 0$.*

Every fuzzy set can be described by $alpha$-cuts.

3.3.2 Knowledge Representation

Fuzzy sets are not enough to deal with more complex structures. To enrich the possibilities of fuzzy sets, they become the basic concepts (*atom*) for more complex elements. The element, which is the next complex one, is the *clause*. It consists of the disjunction of atoms. The conjunction of clauses forms a *formula*. This hierarchy constitutes a very flexible way to attribute different propositions to entities. Within propositional logic, the structure of this formula is called conjunctive normal form, a well-known, powerful concept.

One atom models one vague concept like "big".

Definition 3.3.6 (Atom) *An atom A is a linguistic term, described by a fuzzy set $\mu_A : \Omega \to [0, 1]$.*

Definition 3.3.7 (Negated Atom) *An atom A is negated ($\overline{A}$) by negating its fuzzy set μ_A: $\mu_{\overline{A}} = \overline{\mu_A}$.*

With the use of atoms, we can define literals.

Definition 3.3.8 (Literal) *A literal L is either an atom A, or the negation of A.*

The disjunction of literals results in clauses.

Definition 3.3.9 (Clause) *A clause is a set of literals $L_i, 0 \leq i \leq n$:*

$$C = L_1 \vee L_2 \ldots \vee L_n = L_1, \ldots, L_n.$$

The conjunction of clauses constitutes a formula.

Definition 3.3.10 (Formula) *A set of clauses $C_i, 1 \leq i \leq m$, forms a formula F:*

$$F = C_1 \wedge C_2 \ldots \wedge F_m = C_1, \ldots, C_n.$$

Clauses as well as formulas can be interpreted as fuzzy sets. This is done using unions and intersections:

$$\mu_C = \mu_{L_1} \cup \mu_{L_2} \cup \cdots \cup \mu_{L_n}$$

$$\mu_F = \mu_{C_1} \cap \mu_{C_2} \cap \cdots \cap \mu_{C_m}$$

There is a big advantage in constructing a formula out of clauses and atoms. All components can be addressed individually. This allows changing a formula by adding or removing a vague concept that means an atom, or using different clauses. The definition of union and intersection ensures that it is possible at all time to compute the updated fuzzy set that interprets the whole formula. This approach supports the dynamic aspects of formulas. With these definitions, it is now possible to model very complex propositions and constrains.

Another measure we need is the degree of consistency. The degree of consistency of a fuzzy clause is a measure of how consistent the information contained in the clause are. This is defined in terms of the interpretation of the clause as a fuzzy set:

Definition 3.3.11 (Degree of Consistency of a Clause) *The degree of consistency Con of a clause C is defined as:*

$$\mathit{Con}(C) = \sup_{\omega \in \Omega} \mu_K(\omega).$$

The degree of consistency is 1, if there is at least one value valid without constrains in the fuzzy set interpretation of the clause. The degree of inconsistency is the complement of the degree of consistency:

Definition 3.3.12 (Degree of Inconsistency of a Clause) *The degree of inconsistency Inc of a clause C is defined as:*

$$\mathit{Inc}(K) = 1 - C(K).$$

For the fuzzy formula we do the same:

Definition 3.3.13 (Degree of Consistency of a Formula) *The degree of consistency Con of a formula F is defined as:*

$$\mathit{Con}(F) = sup_{\omega \in \Omega} \mu_F(\omega).$$

The degree of consistency of a formula indicates how compatible the contained clauses are to each other. If there is a at least one value within the interpretation of the formula as a fuzzy set with a degree of 1, the formula is consistent and the degree of consistency is 1. If the contained clauses contradict each other, the degree of consistency for the formula drops. Are the clauses totally inconsistent, the degree of consistency is 0. For the degree of inconsistency for a formula, we have:

Definition 3.3.14 (Degree of Inconsistency of a Formula) *The degree of inconsistency Inc of a formula F is defined as:*

$$Inc(F) = 1 - C(F).$$

3.3.3 Knowledge Processing

There are different methods to process uncertain information. In the field of uncertain implication are two methods prevailing: On the one hand it is possible to specify a fuzzy set by enlarging the information. This can be gained through intersections of different fuzzy sets. The problem related with this approach is that the fuzzy set can become empty in case of inconsistent information. On the other hand there is an approach addressing this problem by keeping two different, inconsistent information. This is done by using the union operator. But problems occur for incomplete information, reaching so far that all elements have a degree of one at the end of the process.

In conclusion, rules are needed on how to process uncertain information. A concept dealing with both of the above mentioned issues takes the uncertain formula as starting point. It defines three important operations on this formula to process information:

Expansion. The expansion adds a new proposition to the formula. This only takes place, if the result is consistent. The operation is called monotone, because the knowledge set grows. As an example, you can add the proposition “hummingbirds are birds” to the knowledge set consisting of the proposition “all birds can fly”.

The γ-expansion for two formulas F_i and F_j is thus defined as:

$$F_i +_\gamma F_j := \begin{cases} \mu_{F_i} \cup \mu_{F_j}, & \text{if } C(\mu_{F_i} \cup \mu_{F_j}) \leq \gamma \\ \mu_{F_i}, & \text{else} \end{cases}$$

Revision. Revision means the addition of a proposition to an existing formula with a consistent result (see also Witte [Wit02b]). The operation is not monotone because it is possible that old information contradicting the new proposition becomes erased. The knowledge set does not grow necessarily. The addition of the proposition "hummingbirds are no birds" to the knowledge set above (containing already the propositions:"hummingbirds are birds", "all birds can fly"), would lead to the erase of the old proposition because of contradiction.

A non-constructive defintion for the γ-revision of two formulas F_i and F_j is:

$$F_i \oplus_\gamma F_j := \begin{cases} F_i +_\gamma F_j & \text{if } C(F_i \cup F_j) \geq \gamma \\ F_i, & \text{if } C(F_j \leq \gamma) \\ F_i' | F_j \subseteq F_i' \subseteq F_i \cup F_j \wedge C(F_i') \geq \gamma & \text{else} \end{cases}$$

(see [Wit02b] for an operational approach).

Contraction. The third operation is the contraction. It removes one proposition from the knowledge set. As with the revision there might not be a definite result. If we have the knowledge set "all birds can fly", "hummingbirds are birds", "hummingbirds can fly" and we want to remove "hummingbirds can fly", it is not enough to remove this single proposition, because it can be inferred from the remaining two.

The γ-contraction for two formulas F_i and F_j is thus defined as:

$$F_i \ominus_\gamma F_j := F_i \oplus_\gamma \overline{F_j}.$$

Defuzzification. As a last step, there is often the requirement of resolving the uncertainty by switching to a non-fuzzy (or *crisp*) representation. This mapping from the fuzzy set representation to precise numbers is called *defuzzification*.

3.4 Summary

Reported speech, beliefs, and fuzzy set theory provide the foundation for this thesis. The order in which we introduced them additionally reflects the chronology for the computational process of the Fuzzy Believer. The core aspect and accomplishment of this work is the successful combination of Natural Language Processing and Fuzzy Set Theory.

For the presentation of our approach in the next chapter, this chapter provided the necessary background.

Chapter 4

Design

In this chapter, we elucidate the design decisions that were taken to model a human newspaper reader starting from the input newspaper articles to the output beliefs of the system.

The task of the system can be described as the simulation of a human newspaper reader, modeling the different ways a human would deal with read information. More precisely, modeling what a human believes after reading a newspaper article. Not every person believes everything he or she has read. Each person has a different background, different knowledge, and different preferences.

Usually the conscious belief process does not start with reading a newspaper article, but already by choosing the newspaper to read. Most newspapers are known to hold certain views on political, economical, or social aspects. This is reflected in the way they report about events. Even the single reporter writing the article can introduce specific subjective opinions about a topic, by choosing different styles of writing and using special language. A very common way in anglophone newspapers is to use reported speech (see Section 3.1) to express opinions. This ensures a more objective approach and limits the influence of the reporter on the article.

Of course, the reporter has to decide whom to cite, thereby highlighting specific views. We decided to limit the possible basic beliefs to the statements encoded in reported speech within a newspaper article. This has a couple of advantages:

- Often, statements reported using reported speech constructs express peculiar or distinctive opinions or beliefs.

- Reported speech statements can be clearly ascribed to a source holding the opinion expressed in the statement.

- Together with *evidential analysis*, it is possible to indicate a degree of reliability into the statement.

The idea of limiting the analysis of newspaper articles to the containing reported speech constructs is not new. A system implementing these ideas was presented in Section 2.2.1.1; it contains a component for extracting reported speech and a profile generation and percolation component. We use this system as a starting point for our Fuzzy Believer system insofar as we also start with extracting reported speech structures. Especially low recall values reported in [Doa03] for the extraction component lead us to the decision to design a new *reported speech extraction* component. The *percolation algorithm* exploiting information gained from evidential analysis and described in [Ger00] could be integrated in our system as part of a belief strategy (see Section 4.5.1.6) We adopt the idea of a *profile generation* identifying same source entities of different reported speech structures. Our design for this component however includes further tasks to be performed apart from this coreference part.

To summarize, the first two tasks our system has to solve are *extracting reported speech*, and *generating profiles*. The core of our system is the processing of the information afterwards. Our focus lies thereby on the analysis of the content of the reported speech utterance and the generation of held beliefs out of it. The system described above is limited to the analysis and processing of the information from the reporting clause, namely source and reporting verb information. Our Fuzzy Believer system can use this information as well but goes beyond this approach by analyzing and processing the reported speech content. The goal is to identify opposite opinions about a topic and accept or reject them according to belief models (*belief strategies*). To tackle this task, we chose *fuzzy set theory*.

Fuzzy set theory explicitly expresses intrinsic fuzziness in natural language, and the handling of ambiguities and similarities in natural languages is done in a more robust way than crisp approaches. Another reason we chose a fuzzy approach are the existing fuzzy operations for representation and reasoning. To incorporate fuzzy set theory, we had to develop new components from scratch to first find a suitable way to transform the statements into a fuzzy representation, then to use fuzzy techniques to process them and subsequently to find the statements the system actually believes. The fuzzy set approach allows a robust processing that can be controlled by various parameters. To process the statements using fuzzy set theory, we need to consider a few constrains: The basic set for fuzzy operations to work on has to be limited to statements dealing with the same topic or fact in the world. This is due to the character of fuzzy processing always considering the whole set to perform its operations on. And with beliefs having nothing to do with each other stored in only one single set, we could not use a similarity measure between statements to perform our computations on, because this would for example lead to the deletion of dissimilar statements dealing with independent topics.

The fuzzy processing task therefore has to consist of four steps:

1. Grouping statements into domains (topics),
2. Finding a fuzzy representation for the statements,
3. Identifying the polarity of statements,
4. Computing beliefs according to a strategy.

For the first step, we use heuristics to compare the predicate-argument structures (PAS) of two statements with each other. To compare only PAS instead of whole sentences is one way to cope with the complexity of natural language. We chose predicate-argument structures as the entities to process because they allow the splitting-up of a sentence into smaller semantic units. The comparison of two PAS is easier than comparing two complex sentences. These PAS are extracted from the output of external parsers and consist of subject, verb, object triples. The fuzzy representation for each statement is the degree of similarity between two statements as computed by heuristics. The polarity finding step is necessary to detect opposing opinions about a topic and is done by evaluating the fuzzy representations of each statement. Different belief strategies are used to model the different ways humans accept or reject beliefs.

The last task to be solved by the system is to find a way to output the defuzzificated results in a clear manner.

The remain of this chapter is structured as follows: We start with an overview of our system and then take a look at the individual system components developed for the Fuzzy Believer system. The components are described chronologically in order of their execution.

4.1 System Overview

In this overview, we present in chronological order the individual processing steps of our system

The starting point of our system is not the selection of newspaper articles, although this could be done, exploiting the huge amount of information the Internet offers. Our system starts with reading in manually chosen newspaper articles, which saves the preprocessing step of finding an article in a web-page, for example [LH02] or [NØ05]. Afterwards, our component-based system starts to process the input document. Different components are used to realize specific tasks within the system, as can be seen in Figure 4.1.

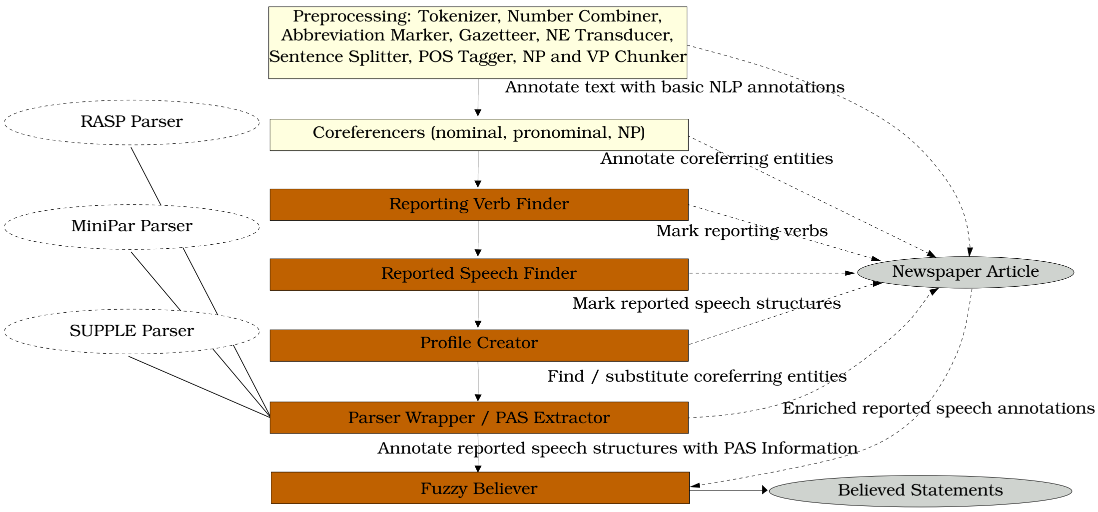


Figure 4.1: Overview of the Fuzzy Believer system components

After preprocessing[1] the input document, a first important step is to identify noun phrases. A noun phrase (NP) is a phrase whose head is a noun or a pronoun. The head can be accompanied by modifiers like determiners, adjectives, or complements. These structures are important to identify acting entities, like persons within a text.

An entity, like a particular person, is usually referred to in different ways within a newspaper article. For the human reader, it is an easy task to recognize that “the President of the United States”, “he”, and “Mr. Bush” are the same person with different designators. Our system should also be capable of identifying the different names for one and the same entity. One of the existing coreferencers (see details in Section C.8) is used for the job.

The next step is to identify reported speech within the document. Therefore patterns have to be found representing the different ways to express reported speech. The next Section 4.2 deals explicitly with this part of the system.

Afterwards we have to combine the found results from the last two steps. The coreference component can identify the same source of two different reported speech utterances. This enables us to build *profiles*. In Section 4.3 we take again a look on this process.

[1] The implementation Chapter 5 gives details about necessary preprocessing.

The most complex part, and an active research field [DGM05, BHDD+06], is to find the *entailment* relation between two sentences. Do they express the same, similar things, contradicting things, or are they totally independent? Our approach uses fuzzy set theory and WordNet to try to decide this question. It is one of the core aspects and gets discussed together with the representation of belief in Section 4.4.

The final step is, after trying to "understand" what has been said and by whom, to define what the system should actually believe. The Fuzzy Believer thus has to do processing on the created belief structure. To model different human "believers", the Fuzzy Believer component uses different strategies. They can be found in Section 4.5.

The result of the system should be a set of propositions the system "believes", and a set of propositions the system rejected. Figure 4.2 gives an impression of the result that the system outputs.

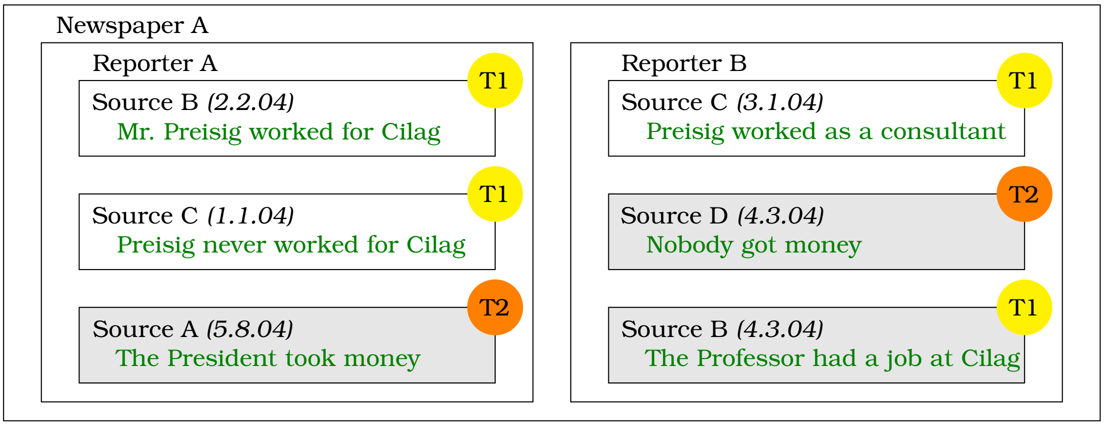


Figure 4.2: System output with different topics T and statements that the system believes (gray background)

4.2 Finding Reported Speech

The starting point for our system is the extraction of reported speech (Section 3.1) from newspaper articles. Finding reported speech in newspaper articles can be done by extracting certain patterns. Especially for the English language, the syntactic rules to use reported speech are very clear. We identified six general patterns to be used for reported speech. They differ in the position of the reporting verb, the source, and the reporting clause. An overview of those six patterns is shown in Table 4.1.

Source	Verb		Content
Verb	Source		Content
Content	Source		Verb
Content	Verb		Source
Content	Source	Verb	Content
Content	Verb	Source	Content

Table 4.1: Six patterns for reported speech in newspaper articles

It is crucial to not only find reported speech sentences as a whole, but to also mark the different elements for further analysis. Our final goal is to get statements in the shape of declarative sentences. This excludes certain reported speech structures whose reported clause is not a grammatically correct statement for its own. Such a construct could be:

The President denied to sign the bill. (4.1)

See Section 3.1.3 on Speech Acts for details. Because our system needs grammatically correct sentences to process, the extracted reported clause has to be a complete sentence on its own. Therefore, we do not consider this kind of reported speech.

There are also reported speech that do not fit into one of our six developed patterns. The following reported speech sentence will only be partially correctly extracted, ignoring the second source and reporting verb.

Mr. Coen predicted that a weak sector in 1987 will be national print – newspapers and magazines – which **he said** will see only a 4.8% increase in advertising expenditures. (4.2)

Constructs that do not fit into our six basic patterns are very rare[2], and additional patterns can be easily added in the future (see Section 7.3).

For our example presented in Chapter 1, this step is visualized in Figure 4.3

The result of this system component is processed by the following component to generate profiles out of the extracted information.

4.3 Generating Profiles

We know from the reported speech structure analysis the sources of the reported speech sentences, but we do not know if they describe the same entity. We do know the newspaper, the article was extracted from, and we know which reporter has written it. In Figure 4.3, we see the extracted sources in italic. We now have to identify the same sources if possible with different degrees of abstraction. For example we could put all "officials" into one group or treat "administration officials" and "law enforcement officials" separately. If we decide to assume a strong identity of the sources, we get for our example the following acting entities:

- law enforcement officials (*Justice Department officials*, *law enforcement officials*, *law enforcement officials*)
- administration officials (*one Reagan administration official*, *Administration officials*, *White House officials*)
- sources (*the sources*, *Reagan administration sources*, *the sources*)
- other officials (*Other officials*)

The existing fuzzy coreferencer component [WB03], which can identify coreferring noun phrases, accomplishes this task. It allows the assignment of the single source occurrences to different entities.

[2] numbers depending on the literary style of the newspaper

North, Meese Often Discussed Contras,
Hostage Release Efforts, Sources Say
By John Walcott

Justice Department officials have said that *Mr. Meese and Col. North regularly met or talked by telephone.*
Adm. Poindexter dismissed the warnings, saying administration lawyers had reviewed Col. North's activities in support of the rebels and had concluded that the Contra aid program was legal, the sources said.
Separately, Reagan administration sources said, *some White House officials became alarmed that Col. North apparently had been collecting and disbursing large amounts of cash for the Contras from his office on the third floor of the Old Executive Office Building next to the White House.* The sources said *Col. North had none of the required government vouchers for the money.*
Nevertheless, Reagan administration sources said *Adm. Poindexter resisted repeated efforts to force Col. North to share responsibility for his secret Iran and Contra operations with other NSC staff members. During the spring of 1986*, the sources said, *NSC specialists in Central American affairs and in intelligence matters separately became concerned about some of Col. North's activities and asked to be included in overseeing them.*
Adm. Poindexter rejected the requests, sources said, *and turned down a suggestion from another aide that the NSC staff's Middle East expert, Dennis Ross, be consulted about the Reagan administration's secret Iran policy.*
"It was clear that Ollie (Col. North) had someone's hand on his shoulder," said one Reagan administration official. *"He was never perceived as an unauthorized loner."*
Administration officials said *they believe Col. North also kept Vice President Bush and his national security adviser, Donald Gregg, informed about some aspects of the Contra aid network.*
Mr. Bush and his aides have tried hard to distance themselves from the scandal, but the team headed by independent counsel Lawrence Walsh has indicated it intends to delve into contacts that Mr. Bush's office had with Contra supporters, according to law enforcement officials.
Interviews with some of Mr. Bush's aides were conducted after the Federal Bureau of Investigation began a full-scale criminal investigation into the matter at the beginning of December, according to law enforcement officials. White House officials said *Mr. Bush was interviewed by FBI agents for 25 minutes on Dec. 12.* Other officials said *they believed a wide-ranging interview with Mr. Bush hasn't been scheduled.*

Figure 4.3: Excerpts from a Wall Street Journal newspaper article from 01.23.87 dealing with Iran-Contra Affair with underlined sources and extracted reported speech in italic

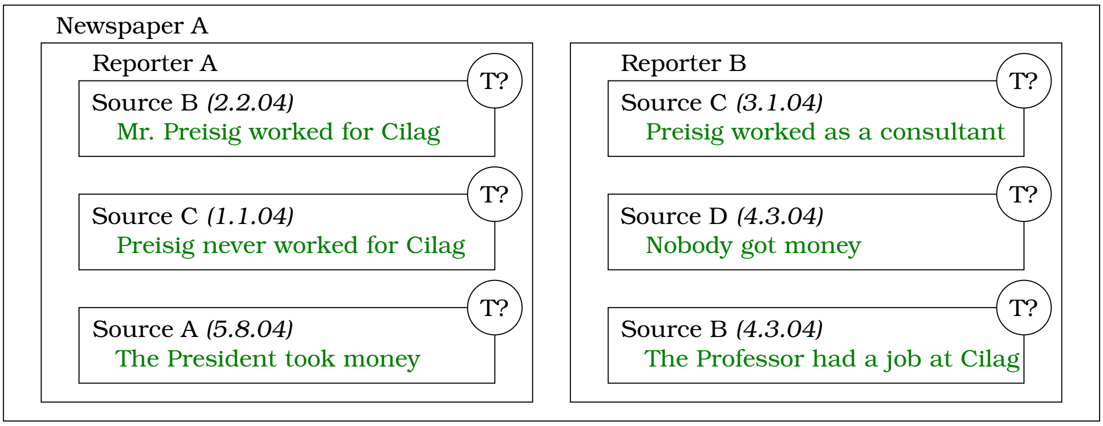


Figure 4.4: The different statements after identifying the source entities, but without topic classification (T?)

4.4 Representing Beliefs

The result of the previous processing steps is a collection of statements of different sources. In this section, we outline the problems related to the representation of these statements together with their sources, and we present our design of a fuzzy representation system. The goal is to create a belief database that can be exploited by further components.

To visualize the information we gained so far and the final results of the system, we have chosen a representation with different boxes similar to [BW91]. To describe the processing step in more detail we take a smaller example. We assume that the system has read 5 different articles from 2 reporters. It has extracted the reported speech information and added the coreference results. This smaller example using the described visualisation is shown in Figure 4.4. Each box contains an extracted reported speech statement together with its source and the date of the article. Additionally, the statements are grouped according to the reporter and the publishing newspaper. Information that is still missing in the figure comprises:

- Identifying the topic (T) of the statement to find an appropriate domain for it, and
- Deciding which statements to believe.

To conclude, the belief base to be generated has to contain different information that is needed later on by different strategies to model an artificial believer:

- The source of a statement:
  1. The person who made the utterance.
  2. The reporter who wrote it down.
  3. The newspaper which published it.
- The date of an utterance.
- The content of this statement.
- The topic of this statement.

- The relationship (similarity, opposition) between two statements.

Apart from the two last points, we have already gathered the required information. But the two remaining points are the most difficult ones for a computational system. For a human, the task to decide if a statement has the same meaning as another statement is rather simple. But for a computer system, this task is most difficult. It involves a lot of knowledge about language and common world knowledge, as well as understanding the syntactical aspect of the statement. The latter part can be handled by state-of-the-art applications, see Section 2.2, but the semantic aspect is still very hard to grasp for computer systems. That makes the decision whether two statements have the same meaning, opposite meanings, or have no relation at all very hard.

If we are able to find out the relations of different statements to each other and save them in our belief base, we will be able to make a trade off between different sources and statements about what to believe.

4.4.1 Fuzzy Belief Representation

To make the difficulties involved in determining the meaning of a statement and the resulting uncertainty observable, we use fuzzy set theory to model our belief base. As stated in Section 3.3.2, we need to define atoms, literals, clauses, and formulas to constitute our model, as well as a meaningful fuzzy interpretation of them. As stated at the beginning of this chapter, the fuzzy set approach causes us to split up our belief data into different *domains*. Each single domain represents a topic, more or less narrowed down to one proposition and their variations. Once a domain has been found for a statement, we have to decide how close the meaning of the statement is to the meaning of other statements within the same domain. Especially interesting are opposing statements that express contrary positions.

The representation of our belief base implies hence two computational steps:

1. Finding the correct domain for a statement.
2. Determining the polarity or the relationship of this statement to other statements within the same domain.

Thus, the processes domain classification and polarity finding have to be separated. Without splitting up the belief base into different topics, the belief revision (see Section 4.5.2.2) for example would drop all statements except for statements of one topic if we choose a similarity measure to build the fuzzy representation.

So before we take a look at the logical foundation of our fuzzy representation and its interpretation, we explain how grouping statements into domains takes place.

4.4.1.1 Domain Finding

One domain contains the statements that have the same or opposite meaning. We try to identify one fact in the world and arrange all statements concerning this fact in one domain. Our example in Figure 4.5 should contain two domains after the classification process:

T1 “Money-taking” and

T2 “Consultant Preisig”.

The different statements are marked with a number identifying the topic, like T1 or T2. Our approach allows a statement to be part of more than one domain simultaneously.

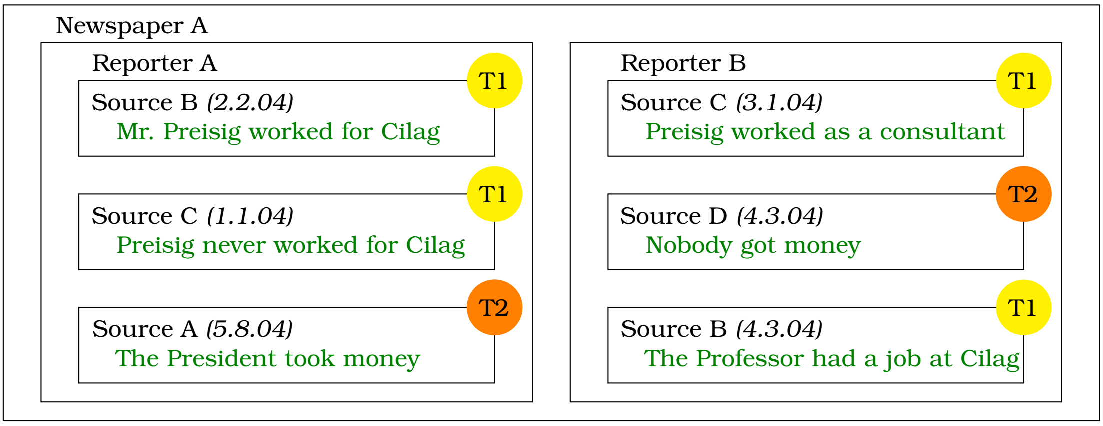


Figure 4.5: The different topics after identifying the domains

Predicate-argument structures. To find statements that belong to one domain, we compare predicate-argument structures of the statements with each other. Predicate-argument structures (PAS) seem to be the core of the semantic structure of all human languages, see [JM00, page 506ff]. The arguments of a predicate are not arbitrary. A verb restricts the usable arguments which can be used with it grammatically and semantically. These small semantic units are well suited for our purpose. PAS are gained by exploiting the output of a semantic parser like RASP, MiniPar, or SUPPLE (see Section 5.2). Different heuristics are used to identify correspondence between two predicate argument structures. We take into account the main verb of a statement together with its subject and one object. Hence, we use the following heuristics to compare two statements:

1. To compare the two subjects:
   - Synonym/Hypernym WordNet (WN).
   - String comparison (ST).
2. To compare the two verbs:
   - Synonym/Hypernym/Antonym WordNet (WN).
3. To compare the two objects:
   - Synonym/Hypernym WordNet (WN).
   - String comparison (ST).

The WordNet (WN) heuristic uses WordNet[3] to find synonyms and hypernyms for a given word. WordNet is based on a lexicon with a hierarchical structure which allows us to define the distance of two words within this hierarchy. We use this distance measure to find similar, related words and give a score to each word pair. The threshold of this score can be adjusted as a run-time parameter, allowing a more lenient or a more strict domain classification. As another run-time parameter, the maximum WordNet distance[4] can be defined. If this parameter is set to 1, only exact synonyms get a score, and statements which use slightly different expressions but have the same meaning would get a 0 score from this heuristic. Especially for the comparison of the two verbs, we rely on WordNet's

[3] www.wordnet.princeton.edu

[4] we use the same WordNet distance as [WB03]

Subject	Verb	Object
preisig	worked	consultant
preisig	worked	consultant
professor	had	cilag
preisig	worked	cilag

Table 4.2: Predicate-argument structures extracted with RASP parser

PAS	Subject Heuristic		Verb Heuristic	Object Heuristic	
	WordNet	String	WordNet	WordNet	String
(preisig, worked, consultant) (preisig, worked, consultant)	0.0	1.0	1.0	1.0	1.0
(professor had, cilag (preisig, worked consultant)	0.0	0.0	0.2	0.0	0.0
(preisig, worked, cilag) (preisig, worked, consultant)	0.0	1.0	1.0	0.0	0.0

Table 4.3: Scores for three pairs of predicate argument structures with maximum semantic WordNet distance of 5.0

semantic distance measure, but other methods are possible as well. The evaluation of different heuristics and their effectiveness is left for future work (see Section 7.3). Note that the component-based structure of our system allows to easily replace old or add new heuristics to make the domain finding process more reliable.

A second heuristic currently in use compares the string representation of two words. This is particularly useful for proper nouns that do not occur within the WordNet database. The score of this heuristic depends on the character overlap of the two words, thus a perfect match is not necessary to gain a score.

To ensure that we compare the appropriate words, an analysis of the main verb is mandatory. We have to differentiate between active and passive constructs, exchanging the syntactic subject and the syntactic object. For future versions (see Section 7.3) a heuristic dealing with temporal aspects of a statement is conceivable. This would require also a temporal analysis of the verb.

Going back to our example from Figure 4.5, Table 4.2 shows the extracted predicate argument structures of the statements for domain T1, and in Table 4.3 you can see the scores of the different heuristics for a few predicate argument structure pairs.

To decide whether two *subjects* are similar enough, the highest score of the subject heuristics is considered. This can be formally described as a disjunction of the single heuristics. The same applies to the *Verb* and the *Object* elements. The requirements for two predicate argument structures to match are that at least two element pairs have at least a matching score of the defined threshold. This threshold can be set as a run-time parameter, and allows for more strict or more lenient domain classification. In our example above (Table 4.3), the following predicate argument structures are part of one domain, if the domain membership threshold is set to 0.5:

- (preisig, was, consultant)
- (preisig, was, consultant)
- (professor, worked, consultant)

This *strict* domain classification demands for a predicate argument structure to be part of one domain to match with all other PAS within the domain. A run-time parameter allows to switch to *non-strict* mode, resulting in larger domains. This more lenient domain classification requires only matching with one predicate argument structure within the domain. This way, we allow transitive relations between the elements of one domain: Consider the following situation, where we have statement 1 with the extracted PAS u, v, and w (s1:u-v-w); a second statement (s2:u-v-x); and a third statement (s3:u-y-x) and a fourth one (s4:z-y-x). S1 and s4 have no PAS element in common, but with the lenient domain classification and s2 and s3 they will be placed in the same domain, because the similarity of s4 and s3 is sufficiently high, the same for s3 and s2, and finally for s2 and s1. For our example, the resulting domain with a threshold of 0.5 is:

- (preisig, knew, consultant)
- (preisig, was, consultant)
- (professor, worked, consultant)
- (preisig, was, consultant)
- (preisig, knew, cilag)
- (preisig, worked, cilag)

Another advantage of dividing the domain classification and the actually matching finding process is that we can use different thresholds for the fuzzy process of assigning statements to different domains and discover supporting and conflicting statements. One statement can belong to more than one domain, exploiting the possibilities of a fuzzy set representation again.

4.4.1.2 Polarity Identification

The next step is the identification of different opinions by detecting the polarity of the statements within one domain. This is the point where we have to dive into fuzzy set theory. We have to find fuzzy representations of the statements and a way to compare these statements with each other.

Logical View. To determine the matching degree of two statements, we use different heuristics H. Each heuristic H_k compares every pair of statements S_i, S_j within one domain with each other. The results are fuzzy atoms $A^{H_k}_{S_{i,j}}$. We use two different kinds of heuristics: positive ones $H_k, k \in [0 \ldots m]$ which indicate a match between two elements , and negative heuristics $H_k, k \in [n \ldots z]$ that exclude a match. Each atom represents one heuristic and a pair of statements and is called a fuzzy literal $L^{H_k}_{S_{i,j}}$. It represents the relation of two statements, S_i and S_j as assessed by the used heuristics H_k. We have so far:

$$A^{H_k}_{S_{i,j}} \quad := \quad H_k(S_i, S_j); \qquad i,j \in [0 \ldots n], k \in [0 \ldots z]$$

which assigns all pairs of statements a fuzzy atom determined by the different heuristics. Each literal $L^{H_k}_{S_{i,j}}$ consists of either a positive atom $A^{H_k}_{S_{i,j}}$ or a negative one $\overline{A^{H_k}_{S_{i,j}}}$ depending whether the heuristic is positive or negative:

$$L^{H_k}_{S_{i,j}} \quad := A^{H_k}_{S_{i,j}}; \qquad i,j \in [0 \ldots n], k \in [0 \ldots m] \cup [n \ldots z]$$

The disjunction of the fuzzy literals for the relation of one statement S_i with another statement S_j describes two clauses; one clause for the positive heuristics using the fuzzy literals gained from the positive atoms, and one clause for the negative heuristics.

$$C_{S_{i,j}^{Hp}} \quad := \bigvee_{j \in [0 \dots n]} L_{S_{i,j}}^{H_k}; \quad i \in [0 \dots n], k \in [0 \dots m]$$

or

$$C_{S_{i,j}^{Hn}} \quad := \bigvee_{j \in [0 \dots n]} L_{S_{i,j}}^{H_k}; \quad i \in [0 \dots n], k \in [n \dots z]$$

This clause contains the different degrees as computed by the heuristics, either positive or negative. The next step is to create fuzzy formulas. The result of the positive and negative heuristics is now combined into this formula:

$$F_{S_{i,j}} \quad := \quad C_{S_{i,j}}^{Hp} \wedge C_{S_{i,j}}^{Hn}$$

If we take a look at the set of formulas $F_{S_{1,j}}$ for example, we get the relation of statement 1 with all other statements in the same domain.

It is possible to use different heuristics to find the relation between two statements. After we have classified our statements into domains, we now need to find possible contradiction or opposing views of the domain's specific matter. Apart from different positive heuristics also anti-heuristics (negative) can be used to compare two statements:

- **Synonym/Hypernym/Antonym WordNet with Negation detection (NEG):** This positive heuristic analyses the verb of the predicate argument structure of a statement. It uses the WordNet data base to find synonyms and hypernyms. The antonym detection together with a basic negation finding approach allows to match statements which contain a negation with statements that contain an antonym of the negated verb.

- **Date/Context verification (CON):** A negative heuristic can be used to identify mismatching dates in the two statements, other numbers or details, or different locations of an event. Consider the following two statements:
  - Sri Lankan planes bombed Tamil rebel strongholds on the Jaffna peninsula in apparent retaliation for a terrorist bombing *Tuesday* of Colombo's main bus terminal.[5]
  - Sri Lankan planes bombed Tamil rebel strongholds on the Jaffna peninsula in apparent retaliation for a terrorist bombing *Monday* of Colombo's main bus terminal.

  Both statements describe a bombing attack, but newspaper agencies might have different information and the two statements do not express the same thing because of a different date. A date/context heuristic could detect such discrepancy and the system can favor one of the two statements.

Fuzzy View. In Figure 4.6 we see atoms expressing the degree of correspondence between S_1 and $S_{1,\dots,5}$, as determined by heuristic H_1.

The interpretation for an atom is the degree of correspondence between one statement with all other statements within the same domain, as determined by one

[5] from WSJ 04.23.87

specified heuristic. Both series of atoms represent positive heuristics. The disjunction of the values of the two heuristics give the fuzzy degrees of the two positive heuristics for a statement and are called a clause. In Figure 4.7 we see the clauses for one statement for the positive heuristics.

If we now combine the clauses for the negative Literals for S_1 with the clauses of the positive literals whose computation we saw in the previous Figure 4.7, we get the formula shown in Figure 4.8. The higher the value obtained for a negative heuristic, the less likely the two statements express the same entity or concept. The resulting formula reflects this fact by assigning a low value to the referring statement. The fuzzy approach allows us to take different degrees of certainty into account. Only if one of the negative heuristics has a value of 100 percent, indicating that the certainty for disagreement of the two statements is at a maximum, the resulting literal will have a value of 0.

4.5 Processing Beliefs

In this section the process of modeling an artificial Fuzzy Believer will be elucidated. The core question for the system is how to interpret the different information collected in the previous steps. To do that, we need to find a model to simulate a human newspaper reader. Different belief strategies are possible and can be implemented. This opens up the possibility to compare the effects of different strategies with each other and to model different human behavior.

Based on our generated belief base, we can now deduce a subset representing the believes of one modelled believer. This subset contains statements chosen out of the belief base according to a particular belief strategy.

To implement the various strategies we make use of fuzzy set theory operations. Before we present these operations, we describe the belief strategies that will be used to model different human newspaper readers.

4.5.1 Belief Selection Strategies

We use the different information gathered in previous processing steps to simulate an artificial believer with certain principles. Different humans have different preferences on what to believe. This aspect should get simulated by the following different basic types of believers:

4.5.1.1 “Believe everything”

This believe type forms the easiest model. Every statement that is identified by the system becomes a part of the belief base of the artificial believer. Opposing or contradicting statements are treated as all other statements and form a part of the beliefs the model holds.

To implement this strategy, we process our fuzzy input by expanding the existing belief base with new incoming statements. For the threshold of the expand-operation we choose 0.0, allowing the belief base to include *all* new statements. The result for our example is shown in Figure 4.9.

4.5.1.2 “Believe old statements”

This strategy prefers older beliefs. New statements are only added to the belief base if they are not opposing existing statements. This model exemplifies a conservative

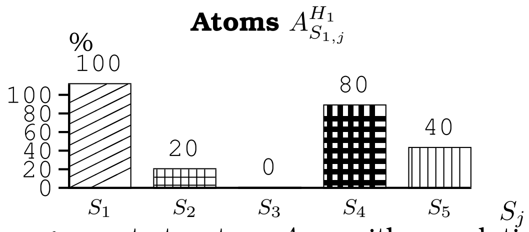
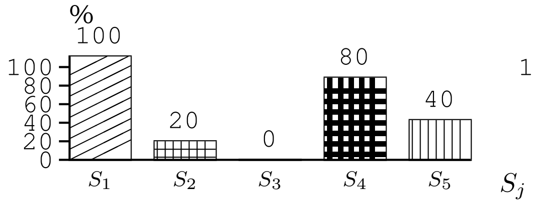
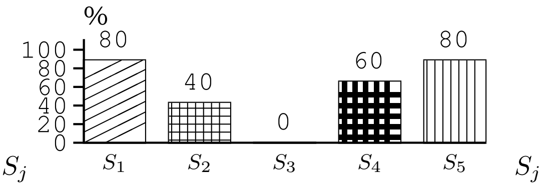


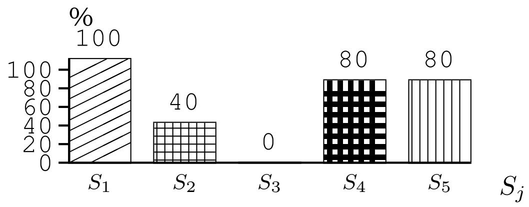
Figure 4.6: Predicate-argument structure $A_{S_{1,j}}$ with correlation grades for all statements in the domain ($S_1, \ldots, S_5$) as computed by heuristic H_1



(a) **Literals** $L_{S_{1,j}}^{H_1}$

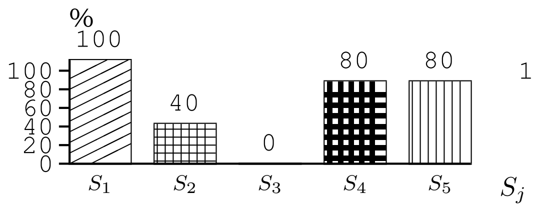


(b) **Literals** $L_{S_{1,j}}^{H_2}$

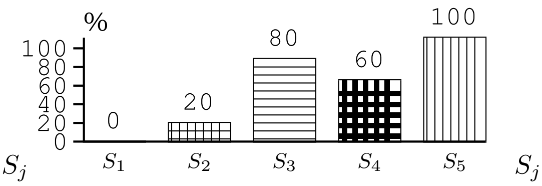


(c) **Clauses** $C_{S_{1,j}}^{H}p$

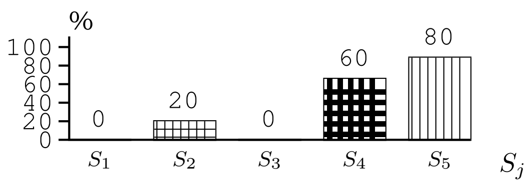
Figure 4.7: Clauses $C_{S_{1,j}}$ with correlation grades for all PAS in the domain ($S_1, \ldots, S_5$) as computed by the positive heuristics H_1 and H_2, whose literals are shown on top



(a) **Clauses** $H_{S_{1,j}}^{Hp}$



(b) **Clauses** $C_{S_{1,j}}^{Hn}$



(c) **Formulas** $F_{S_{1,j}}$

Figure 4.8: Formulas $F_{S_{1,j}}$ as resulted from combining the clauses on top, $C_{S_{1,j}}^{H}n$ and $C_{S_{1,j}}^{H}p$

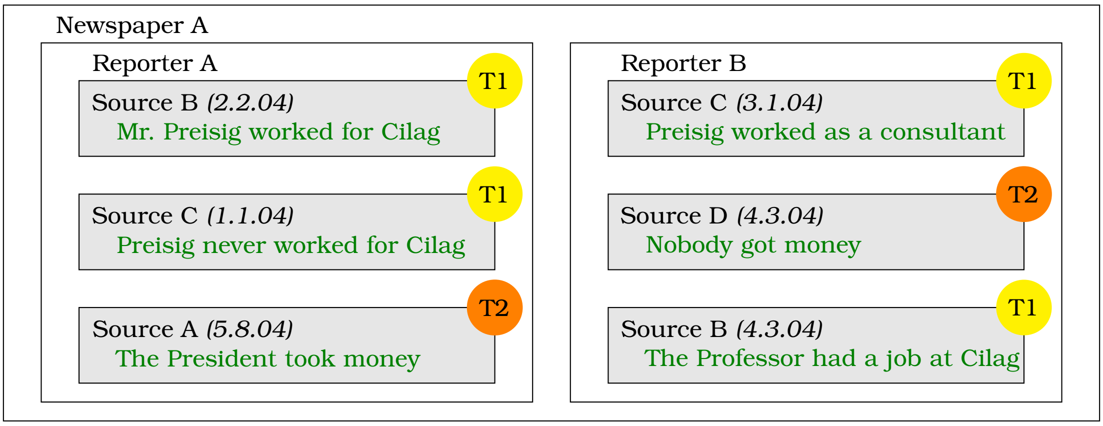


Figure 4.9: Believe Everything: The system believes gray statements

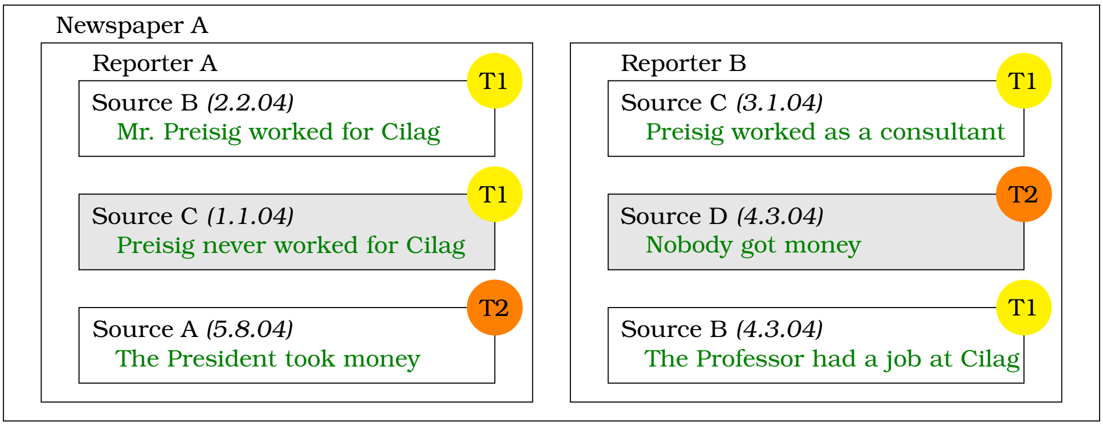


Figure 4.10: Believe Old Statements: The system believes gray statements

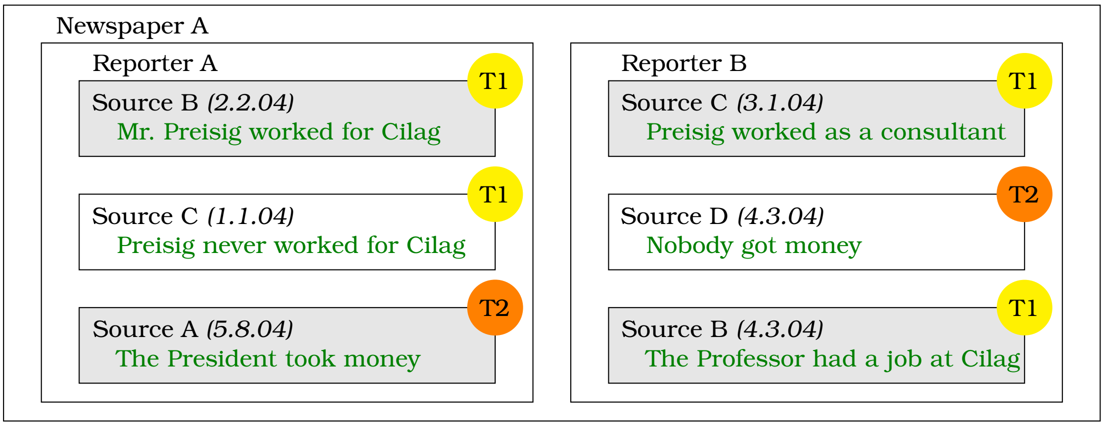


Figure 4.11: Believe new statements: The system believes gray statements

point of view. New statements that conflict with old ones are rejected. The result for our example is shown in Figure 4.10.

We can use the expand-operation again to model this type. Here, the threshold value has to be higher than 0.0. The threshold determines the degree of agreement between an old statement and a new one. A value of 1.0 means that only new statements that have exactly the same meaning as the old statements in the belief base are added to the belief base. A value less than 1.0 also allows statements to be added that have similar meanings.

4.5.1.3 "Believe new statements"

The opposing strategy to the one above prefers new statements. That means if a statement is found that has an opposing meaning to an old statement within the belief base, the old statement gets removed and the new one gets added.

We can use the revision formalism on our fuzzy sets to implement this strategy. As long as the ordering of the revision on the statements is done according to the date of the statement, the revision results in preferring new statements over old opposing ones. The outcome of this strategy for our example can be seen in Figure 4.11.

4.5.1.4 "Believe in the majority"

A more realistic model counts the number of the different point of views concerning one statement and chooses the one the majority holds.

This can be implemented using the or-operation. For each statement within one domain we try to merge this statement with other statements of the same domain. The merging process tries to find statements with the same meaning and puts them together. The result are one or more sets of statements representing one meaning. Our believer chooses then the meaning which is supported by the most statements in the domain. In Figure 4.12 this strategy is applied for our example.

4.5.1.5 "Believe in a certain newspaper/reporter/source"

Instead of looking at the majority, a believer could trust a special newspaper, reporter, or source. Different newspapers are known to have certain political ten-

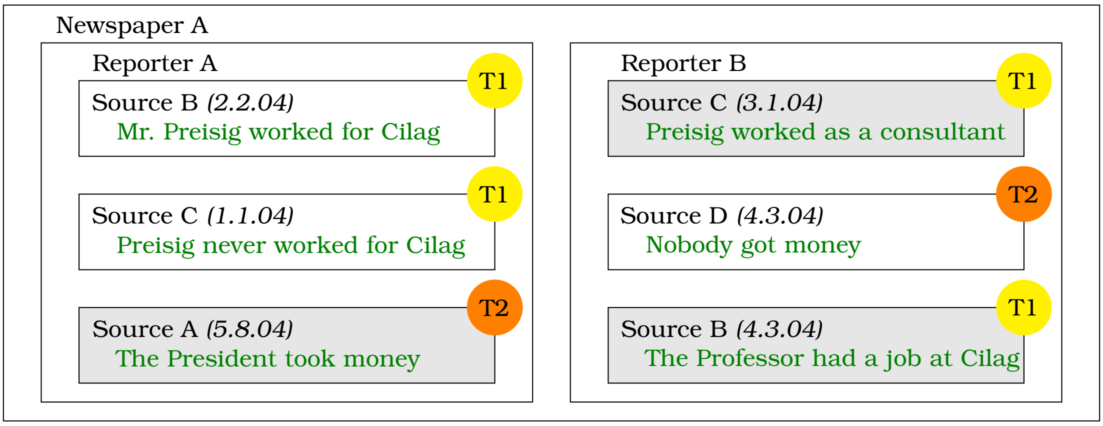


Figure 4.12: Believe majority: The system believes gray statements

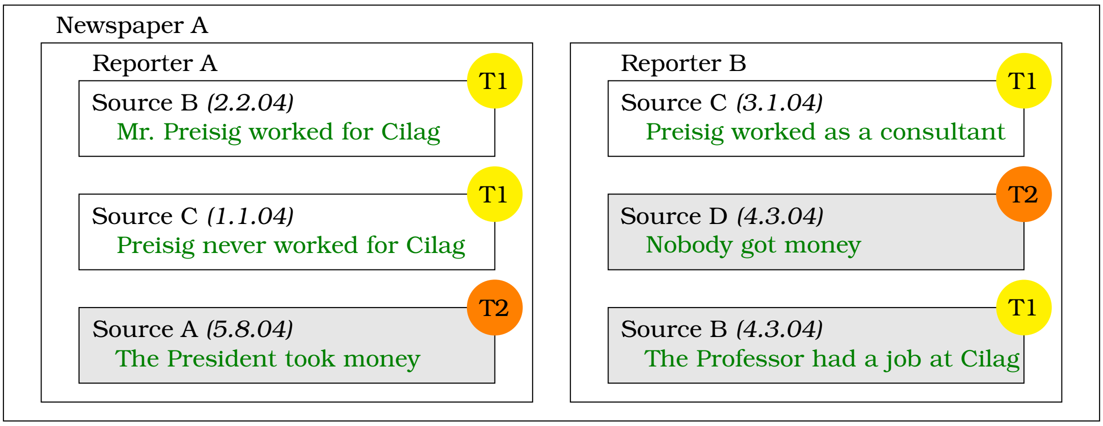


Figure 4.13: Believe certain newspaper/reporter/source

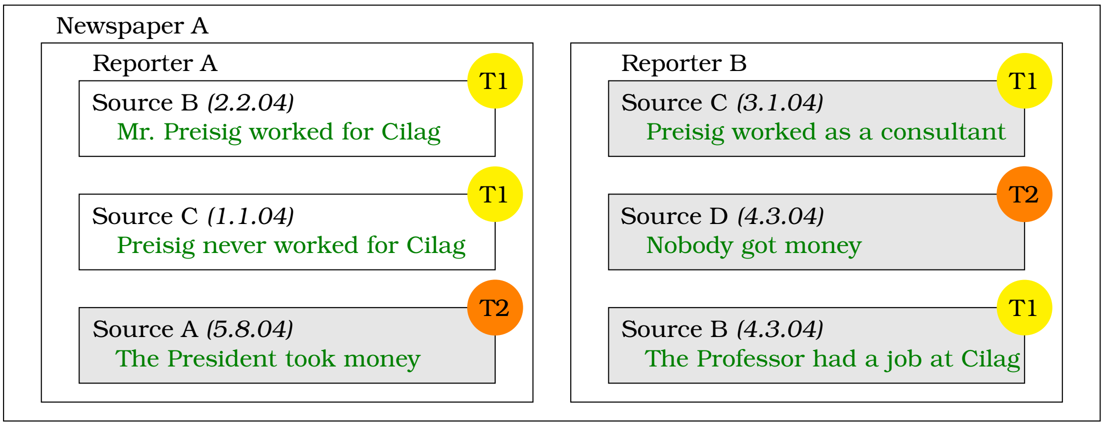


Figure 4.14: Believe weighted majority: The system believes gray statements

dencies, representing special opinions. An artificial believer can also give higher credibility to certain sources, for example an official police spokesman might be more trustworthy than a second rate witness of an incident. The result for our example is shown in Figure 4.13.

The technical difficulty is to find statements from the same source. This can be solved by using the extracted coreferences.

4.5.1.6 "Believe in weighted majority"

A more complex model taking both newspaper/reporter/source and majority into account is the weighted majority strategy. We get this by combining the two belief selection strategies above. One possible result with different confidence into different sources is shown in Figure 4.14. Different formulas are possible to include the degree of confidence in a newspaper, reporter, and source and the number of occurrence of a particular statement. For a deeper analysis of the reliability of a statement see [Ger00] page 67ff. We propose the following formula to compute a level of certainty (LOC) into the common meaning m_i of statements $s_{i1}, \ldots, s_{ik}$:

$$\mathrm{LOC}(m_i) = \sum_{j=1}^{k} \frac{1}{3} * (C_{sys}(np(s_{ij})) * C_{np}(rep(s_{ij})) + C_{sys}(rep(s_{ij})) * C_{rep}(src(s_{ij})) + C_{sys}(src(s_{ij})) * C_{src}(s_{ij})) \tag{4.3}$$

with C_{sys} the confidence degree of the system, C_{np} the confidence of the newspaper, C_{src} the confidence of the source, and C_{rep} the confidence of the reporter. The values have to be between 0.0 and 1.0. The confidence of the newspaper into its reporters should be high, but a newspaper can publish articles from reporters without having 100% confidence in them. For a newspaper reader this confidence degree is hard to grasp. The confidence of a reporter into a source can often be inferred from the way the reporter introduces the source, and how he describes and relates to it. The confidence of a source into a statement might be encoded in the circumstantial information or the chosen reported verb [Ber92], see also Section 3.1.4.

Other formulas to compute a level of certainty are possible and maybe more realistic. In [Ger00] a similar formula to compute a level of certainty is presented. It does not take the newspaper the article is published in into account, but contains the other information we also consider:

$$\mathrm{LOC}(s_i) = C_{sys}(src(s_{ij})) + C_{sys}(rep(s_{ij})) * C_{rep}(s_{ij}) + C_{sys}(src(s_{ij})) * (C_{src}(s_{ij}) + C_{rep}(src(s_{ij}))) \tag{4.4}$$

The single values can be 1.5 (high), 1.0 (neutral), or 0.5 (low). In addition to our formula, this one contains the degree of confidence of a reporter into a statement. We encode the same information as confidence of the source into the statement. It depends on the individual context to whom one should attribute this information: Sometimes the reported verb mirrors the way the source talks and sometimes the reported verb indicates the feelings of the reporter towards a statement.

Coming back to our proposed formula, it should be made clear that it is only a first approach that can function as a baseline for future implementations. To get an actual system result, we have to do some more processing after computing a LOC for all merged statements. Our artificial believer can now choose to believe the merged statements with the highest level of certainty, or, alternatively, it is also possible to define a threshold for a LOC, making *potential beliefs* to *held beliefs* if the value crosses the defined threshold. The notions potential and held beliefs were defined in [Ger00] page 61f, and gives the system the ability to discriminate between statements uttered by many people or people with a high degree of confidence in them, and statements that are maybe only uttered once without further support or contradiction.

4.5.2 Fuzzy Belief Processing

Now that we gathered all the statements, clustered them into domains, and computed the correspondence between the single statements within one domain, we will use this information to process the statements according to different believe models. We use different fuzzy operations to accomplish this goal.

4.5.2.1 Expansion

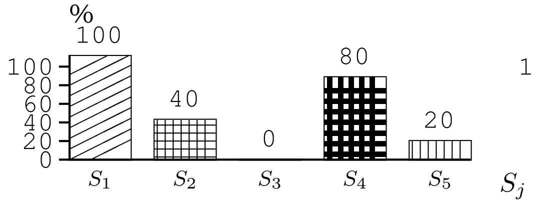
To implement the "believe all" and "believe old news" strategies, we will need a fuzzy operation called *expansion*. To explain expansion we have to go back to the logical level of our representation. On a logical level, this means we have to do a γ-expansion: $+_\gamma$, where γ is the degree of consistency that has to be achieved after the operation to allow the expansion. An example for the γ-expansion with $\gamma = 0.8$ is shown in Figure 4.15. The new fuzzy set formula $\mu_{F_{S_2,S_4}}$ represents now the correspondence of S_2 and S_4 with a consistency degree of at least 0.8.

Because the expansion of a formula 1 with a formula 2 is not the same as an expansion of formula 2 with formula 1, we have to pay attention of the ordering in which we execute the expansion. We decided to order the formulas according to the date, the statements represented in the formula were uttered. For a first approach we use the date of the occurrence of the article in the newspaper, but a more detailed analysis of circumstantial information could give different dates for statements. Example (4.5), taken from WSJ Monday 04.06.87, exemplifies this fact:

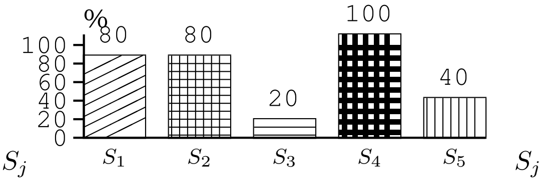
> However, Goodyear said **Friday** that discussions with prospective buyers are continuing, but there hasn't been a concrete agreement. (4.5)

4.5.2.2 Revision

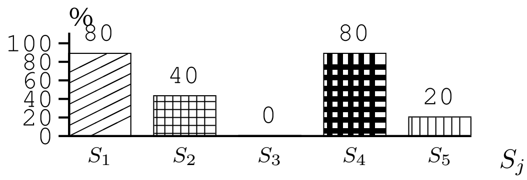
The "believe new news" strategie requires another fuzzy operation. For this task, we use *revision*. The result of a revision of formula 1 with another formula 2 depends on the order of the formulas, as well as on an ordering of the clauses of the formulas. The ordering can be chronological depending on the timestamp of the insertion



(a) **Formula** F_{S_1}

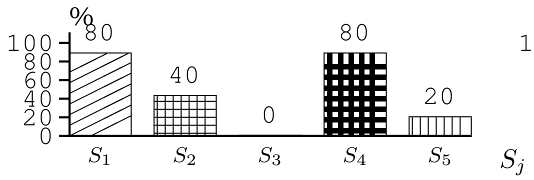


(b) **Formula** F_{S_4}

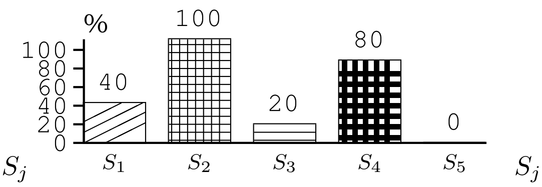


(c) **Formula** $F_{S_4} +_\gamma F_{S_2}$

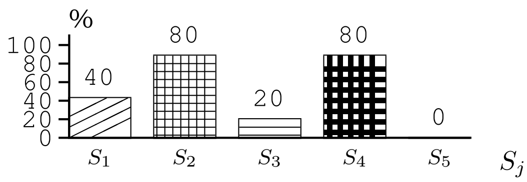
Figure 4.15: Result of γ-expansion with $\gamma = 0.8$ of the two formulas on top



(a) **Formula** $F_{S_4} +_\gamma F_{S_2}$



(b) **Formula** F_{S_2}



(c) **Formula** $(F_{S_1} +_\gamma F_{S_4}) \oplus_\gamma F_{S_2}$

Figure 4.16: Result of γ-revision with $\gamma = 0.8$ of the two formulas on top

of the clause into the formula, or any other ordering like ordering according to degrees of certainty or an order relying on the reporter or newspaper. We chose as an order the first way enabeling as to model a belief strategy concerned with the chronological order of news.

The revision process compares clauses and if the γ-expansion of the formula 2 with a clause of the formula 1 is successful, this single clause gets added to the formula 2. Clauses from formula 1 that are not consistent with formula 2 are rejected. In Figure 4.16 we see the formula generated in the previous step containing two clauses (Figure 4.16(a)), and next to it the new formula, with which we start the revision. The result shown in Figure 4.16(c) is a new formula containing two clauses. The ordering of the formulas which determines the sequence of processing is again defined by the date of the statements.

```
VECTOR DOMAIN.MERGE_OPINIONS( Vector opinions, degree y )
begin
 Opinions o1, o2 ;
 Vector mergedOpinions := new Vector();

 while (opinions.size() > 0) do
  o1 := opinions.remove[1];

  // check if the opinion supports or contradicts the already merged opinions
  for (Iterator i := mergedOpinions.iterator(); i.hasNext (); ) do
   o2 := i.next ();
   if (o1.merge(o2, y))

    // Opinion o2 was merged with o1 and can get removed
    then i.remove ();
   fi
  od

  // check the remaining opinions
  for (Iterator i := chains.iterator (); i.hasNext (); ) do
   o2 := i.next ();
   if (o1.merge(o2, y))

    // Opinion o2 was merged with o1 and can get removed
    then i.remove ();
   fi
  od
 mergedOpinions.add (o1 );
 od
 return(mergedOpinions);
end
```

Figure 4.17: Algorithm used for merging statements to different opinions within one domain

4.5.2.3 Merging

To implement the "believe majority", and also the "believe weighted majority" strategies we need to count the statements belonging to the opposing and supporting opinion groups. Therefore we need to group these statements according to their polarity. The result are different groups containing statements expressing different opinions about a topic. It is now possible to count the number of statements for each group and prefer the opinion with the most support.

Merging two formulas is similar to expansion. The merging is successful, if the expansion of the two formulas would be successful. But the result is not the result of the expansion with the conjunction of the formulas, but a disjunction of the containing formulas. The algorithm used for merging is taken from [Wit02a] and is shown in Figure 4.17.

4.6 Summary

We have presented the design for the main components of our Fuzzy Believer system. For the first two task – the extraction of reported speech from newspaper articles, and the finding of coreferring sources – we could orientate us at existing systems [Doa03, Ger00]. Anyhow we had to develop new grammar rules to extract reported speech and a new implementation of the profile generation task that can deal with our data representation.

For the analysis of the reported speech content we had to come up with a totally new design, solving problems of belief representation and belief processing. We chose fuzzy set theory as a robust way to deal with uncertainty in natural language processing. We have identified individual subtasks that have to be solved for the processing step:

- Grouping statements into domains (topics),
- Finding a fuzzy representation for the statements identifying their polarity,
- Computing beliefs according to a strategy.

In the following chapter, we will describe the implementation of these tasks in detail and how they depend on each other.

Chapter 5

Implementation

In this chapter, we describe the implementation of our system, which is based on the GATE environment. It utilizes data structures, classes and standard components of GATE, as well as modified and self-developed components. Therefore, the first part of this section briefly discusses GATE and its concepts, and the second part describes details of the existing system's implementation.

5.1 GATE Environment

GATE [CMBT02] is the abbreviation for General Architecture for Text Engineering. As stated in the name, it offers an architecture for language processing software, by providing a framework implementing the architecture. It has been in development at the University of Sheffield since 1995 and has been employed in a lot of research projects. In addition to the architecture, GATE includes a development environment built on top of the framework. It contains graphical tools for editing and developing new components as well as tools to visualize generated results.

GATE is based on the Java language, and can be run on any computer platform. Because it uses Unicode [Con03], it can be used for a great variety of languages. The open-source aspect – GATE is distributed under GNU Library General Public License[1] and is freely available for creating applications – and the intuitive handling made it popular within the research community.

5.1.1 GATE Overview

We give a short overview of the GATE framework, details can be found in [CMB+06]. GATE components are known as *resources*. They are reusable software chunks with well-defined interfaces and of a popular architectural form, used in Sun's Java Beans and Microsoft's .Net. A specialized type of Java Bean constitutes a GATE component.

Each component can be divided into three main categories:

Language resources: Include mostly textual entities such as documents, thesauri, lexicons, corpora or ontologies;

Processing resources: Include algorithmic entities like parsers, sentence splitters, regular expressions processors;

Visual resources: Represent visualization and editing components that are used in the GATE graphical user interface.

[1] http://gate.ac.uk/gate/licence.html

The set of resources integrated in GATE is called CREOLE (a Collection of REusable Objects for Language Engineering). All the resources within CREOLE are packed as Java Archives (JARs), plus some XML-files for configuration purposes.

5.1.1.1 Language resources

The most important language resources in GATE are *Corpus*, *Document*, and *Annotation*. They form the main input and output of the natural language processing system, whereas each Corpus consists of a set of Documents and a Document itself consists of text to be processed and a set of Annotations. Annotations can be arranged to Annotation sets, making it simpler to locate and manage them.

An Annotation has always a name, specifying its kind, and may have *Features*: a set of *feature_name* = *feature_value* pairs. Features describe properties that may be assigned to the text passage bounded by the Annotation.

5.1.1.2 Processing resources

There is a number of standard Processing Resources in GATE, which process natural language text either generally or process it for building up a level of relatively low and universal textual structures (e.g., parts-of-speech), and so, such resources may be used in any application independent of a domain. We present in the next section a description of the Processing Resources used within our Fuzzy Believer system. Most of them are part of the ANNIE ("A Nearly-New Information Extraction") System shipped with GATE and were adapted for the use in our system.

Before we go into detail of specific components, we briefly introduce a tool widely used within GATE to process text data:

JAPE Transducer. JAPE stands for Java Annotation Patterns Engine and provides finite state transduction over text annotations based on regular expressions [CMT00]. A grammar written in JAPE is compiled into a transducer It consists of a set of phases, each of which consists of a set of pattern/action rules. The pattern (the Left-hand-side) of a rule is an annotation pattern that may contain one or more regular expression operators out of:

- *Zero or more* (*) operator,
- *Alternation* (|) operator,
- *Optional* (?) operator and
- *One or more* (+) operator.

The action (the Right-hand-side) consists of annotation manipulation statements, which may be Java code, referring to the GATE framework classes and to annotation labels matched on the Left-hand-side.

The Jape Transducer has a *Control* option, which specifies the method of rule matching for the case, when several rules can be fired at a certain text segment. The *Input* option specifies which annotations will be considered by the transducer during the pattern matching process. The Jape Transducer supports *label-binding* schemes. Labels are specified in a pattern, and bind to the last matched annotation in its scope. Annotation Sets, which are the sets of Annotations matched by the label when the rule fired, can be used in an action code.

5.1.1.3 GATE applications

A user application running in the GATE environment usually consists of:

- A corpus, i.e., a set of documents, being processed
- A set of Processing resources, standard or implemented as CREOLE by a user
- A set of Visual resources for extended text processing being run in the GATE graphical user interface

The Processing resources run in a predefined sequence defined by ordering of the components in a *Pipeline*. Input-output connections are thereby structured, and the results of work of one resource may be used by the following one, by adding *annotations* to the document being processed by the pipeline.

5.1.2 Components used in the Fuzzy Believer System

To implement the Fuzzy Believer, a set of components is necessary. Some of those components are shipped with GATE and can be used with minor adoptions for our system. In the below list you can distinguish by their name components from GATE/ANNIE, components from CLaC[2] (both GATE/ANNIE components with minor changes and CLaC-developed components), and components developed by IPD[3]. The component names written in italic indicates omponents developed by ourselfs as part of the Fuzzy Believer System.

The components used – see also Figure 4.1 – in order of their execution, are:

CLaC Document Resetter: If one text is being processed more than one time, this component ensures that the previous results for this text is removed so no interference with new results are possible.

ANNIE Tokenizer: The tokenizer identifies words, space-tokens, numbers, and punctuation as well as other symbols.

CLaC Number Combiner/Interpreter: This component combines the extracted numbers and symbols like “4”, “.”, “6”, and “cm” to “6.4cm”, and interprets words like “one” as “1”.

CLaC Abbreviation and Acronym Marker: This marker goes through the text and compares it with a list of abbreviations. If it finds a corresponding pair, it annotates it as abbreviation and adds the complete expression as a feature.

ANNIE Gazetteer: The Gazetteer adds an annotation to phrases found in different lists and gives them major type and minor type features.

CLaC Named Entity Transducer: A multi-stage JAPE transducer identifies several named entities, like Persons, Organizations, Locations, or Number and Date information.

CLaC Sentence Splitter: This component divides the text into sentences trying to identify the start of a new sentence by considering abbreviations and tokens.

[2] Computational Linguistics at Concordia (CLaC), Department of Computer Science, Concordia University, Montréal, Canada

[3] Institute for Program Structures and Data Organization (IPD), University of Karlsruhe, Germany

ANNIE POS Tagger: Part-of-speech tagging is performed by the Hepple tagger [Hep00] included in the GATE distribution.

GATE Verb Phrase Chunker: This transducer implemented in JAPE, identifies verb groups and annotates them with tense, voice, type, etc. information.

IPD Multilingual Noun Phrase Extractor: The Noun Phrase Extractor (MuNPEx)[4] is a base NP chunker, i.e., it does not deal with any kind of conjunctions, appositions, or PP-attachments. It is implemented as a JAPE transducer and can make use of previously detected named entities (NEs) to improve chunking performance.

CLaC Max NP Transducer: This Component tries to maximize the previously found noun phrases by including appositions, conjunctions, etc.

GATE Morphological Analyser: Considering one token and its part-of-speech tag, one at a time, the Morphological Analyser identifies its lemma and an affix. The token annotation features is extended with the two new items.

CLaC Reporting Verb Finder: A manually generated list of reporting verbs is transduced and if one of them appears in the text, it is annotated as *Reporting Verb*. The annotation also contains semantic information about the particular reporting speech verb, like polarity, strength, explicitness, etc. Details for the implementation of this component can be found in 5.3.1.

***Reported Speech Finder*:** This multi-stage JAPE transducer uses a set of patterns to identify reported speech within a text. It also annotates the found sentences with *source*, *verb*, and *content* features. Details for the implementation of this component can be found in 5.3.2, the design is described in Section 4.2

ANNIE Orthographic Matcher: This Gate component identifies relations between named entities and adds this information to the corresponding NE-annotations.

ANNIE Nominal Coreferencer: The Nominal Coreferencer tries to match "`JobTitle`"-annotations with "`Person`"-annotations.

ANNIE Pronominal Coreferencer: This component uses different modules to identify the antecedence of a pronoun.

IPD Fuzzy Coreferencer: This component builds fuzzy coreference chains [WB03] using different heuristics to find coreferences.

***Profile Creator*:** The Profile Creator (for the design see Section 4.3) assembles all information of the previous steps necessary to build up reported speech profiles. Especially the coreferences of the source is searched to identify same sources. Details for the implementation of this component can be found in 5.3.3.

***Parser Wrapper & PAS Extractor*:** Different Parsers can be used to extract a predicate argument structure of the reported speech content (see Section 4.4.1.1). Details for the implementation of this component can be found in 5.3.4.

***Fuzzy Believer*:** This component contains the Fuzzy Believer framework. It implements different belief strategies to simulate a human reader Details for the implementation of this component can be found in 5.3.5, the design aspects are described in Section 4.4and Section 4.5.

[4] Multi-Lingual NP Chunker (MuNPEx), `www.ipd.uka.de/~durm/tm/munpex/`

The most important components within the Fuzzy Believer Application are described in more detail below. Most of the implementation details about the ANNIE/-GATE components can be found in [CMB+06].

5.2 Parsers

Apart from the components that can be integrated in GATE, we also rely on external Resources, namely natural language parsers. These parsers have a crucial function within the system as a whole. Their output is used by the Fuzzy Believer subsystem to make different decisions, as will be seen later in Sections 5.3.5.1 and Section 5.3.5.3.

5.2.1 SUPPLE

SUPPLE [GHS+05] is a natural language parsing system implemented in Prolog. It can be run as a stand-alone system or integrated in the GATE framework. SUPPLE produces syntactic as well as semantic annotations to given sentences.

It is a general purpose bottom-up chart parser for feature-based context free phrase structure grammars. Grammars are applied in series allowing to choose the best parse for each step and continue to the next layer of grammar with only the selected best parse. This leads to a non-complete output on the one hand, but on the other hand speeds up the parsing process a lot. Furthermore, it allows the development of sub-grammars more easily.

The identification of verbal arguments and attachment of nominal and verbal post-modifiers, such as prepositional phrases and relative clauses, is done conservatively. Instead of producing all possible analyses or using probabilities to generate the most likely analysis, SUPPLE only offers a single analysis that spans the input sentence only if it can be relied on to be correct, so that in many cases only partial analyses are produced.

5.2.2 MiniPar

MiniPar is a broad-coverage English parser. It uses a manually constructed grammar and is like its predecessor PrinciPar [Lin94] a principle-based parser, instead of a rule-based one. The grammar is represented by a network with grammatical categories as nodes and the syntactic relations as links. In total, the grammar contains 35 nodes and 59 links, as well as dynamically created nodes and links for subcategories of verbs. The algorithm is a distributed chart parsing algorithm implemented as a message passing system. Each node has its own chart containing partially built structures referring to its representing grammatical category. The grammatical principles are constraints associated with the nodes and links. MiniPar uses a derived form of WordNet [Fel98] as lexicon with some additional entries. Each entry in the lexicon consists of a word together with its possible part-of-speech tags. The ambiguities in the lexicon are dealt with by the parser instead of an external tagger. Although MiniPar uses a constituency grammar for internal processing, the output is a dependency tree showing the category of each word and its dependency (grammatical) relations. These relations consist of a word called *head*, and another word called *modifier*.

For a more detailed evaluation of this parser see [Lin98].

5.2.3 RASP

RASP [BCW06] stands for "robust accurate statistical parsing" and is a parsing system for English language documents. It consists not only of the actual parser itself, but also contains modules for the required preprocessing steps, like tokenization, part-of-speech tagging, sentence splitting, and lemmatization. This enables the parser to be run stand-alone, but replicates the work in a framework like GATE, where the low-level information has already been generated.

A non-deterministic LALR(1) table is constructed automatically using a feature-based grammar. Out of this LR table, a parse forest is generated and different trees are assigned a probability. The n-best parse trees are subsequently ranked using probabilistic methods.

The system has been evaluated using 560 manually annotated sentences from the Wall Street Journal and achieved 78 percent precision and 75 percent recall for the grammatical relations extraction. For details on the evaluation of RASP, see [BC06].

5.3 Fuzzy Believer System Components

After describing standard components used within the Fuzzy Believer system, this section gives a deeper insight into the special components developed for the Fuzzy Believer. The first two use JAPE grammars to extract features from the document. The other three are implemented in Java making use of the Gate component framework.

5.3.1 Reported Verb Finder

The reported verb finder is implemented in JAPE. It consists of a set of rules to find verbs used to express reported speech. Therefore, it mainly makes use of the annotation added by the ANNIE verb phrase chunker (see Appendix C.6). This component was first developed and implemented by Doandes [Doa03] to extract information related to evidential analysis. A few verbs were removed and others added, to improve precision and recall. Especially verbs that are also likely to occur in non-reported speech situations like "agree", "deny", or "decline", have been treated specially, by ensuring that there is a reported speech structure, indicated for example by ":" or "that" after the reporting verb. This action on the one hand limits the recall but on the other hand boosts the precision, leading to a better overall performance.

A complete list of verbs used to identify reported speech sentences can be seen in Table 5.1. If the system detects one of these verbs, it is possible that the containing sentence is a reported speech structure. In Figure 5.1, a code snippet can be seen showing a JAPE-rule for the verb "concede". And after detecting these verbs within a document, the reported verb finder marks them as reported speech verbs by adding an annotation containing the root of the main verb of the reported speech verb phrase. In addition, it also adds further information about semantic dimensions [Ber93] that can be gained from the specific verb. These comprise: voice, explicitness, formality, audience, polarity, presuppostion, speech act, affectedness, and strength. These information differ from verb to verb. They can be used later to determine, for example, the confidence of the reporter into the source. These information are not yet evaluated by the fuzzy believer (see future work Section 7.3), but were a main aspect in Doandes [Doa03] to implement a system for evidential

according	accuse	acknowledge	add
admit	agree	allege	announce
argue		assert	believe
blame	charge	cite	claim
complain	concede	conclude	confirm
contend	criticize	declare	decline
deny	describe	disagree	disclose
estimate	explain	fear	hope
insist	maintain	mention	note
order	predict	promise	recall
recommend	reply	report	say
state	stress	suggest	tell
testify	think	urge	warn
worry	write	observe	

Table 5.1: Reporting verbs identified by the reported verb finder

```
Rule: concede
(
 {VG.voice == "active", VG.MAINVERB == "concede"}
):rvg -->
{
 gate.AnnotationSet rvgAnn = (gate.AnnotationSet)bindings.get("rvg");
 gate.AnnotationSet vgclac = rvgAnn.get("VG");
 gate.Annotation vgclacAnn = (gate.Annotation)vgclac.iterator().next();
 gate.FeatureMap features = Factory.newFeatureMap() ;
 features.put("mainverb", vgclacAnn.getFeatures().get("MAINVERB"));
 features.put("semanticDimensions","VOICE:unmarked EXPLICITNESS:explicit
   FORMALITY:unmarked AUDIENCE:unmarked POLARITY:positive
   PRESUPPOSITION:presupposed SPEECH_ACT:inform AFFECTEDNESS:negative
   STRENGTH:unmarked");
 outputAS.add(rvgAnn.firstNode(),rvgAnn.lastNode(),"RVG",features);
}
```

Figure 5.1: A sample for the JAPE grammar to find and mark reported speech verbs

analysis (see 3.1.4). We enriched our reported verb finder incorporating the work of Doandes on semantic features of verbs. Currently, we only make use of the identification of the reported speech verb, to use it for the reported speech finder, which is described in the next section.

5.3.2 Reported Speech Finder

The reported speech finder is implemented in an enhanced version of the JAPE language using the Montreal Transducer[5]. The grammar consists of a set of rules to identify reported speech structures. One of the six rules can be seen in Figure 5.2. This rule detects for example the following reported speech structure and marks the source, reported verb, and the content[6]:

[5]Developed by Luc Plamondon, Université de Montréal, http://www.iro.umontreal.ca/~plamondl/mtltransducer/

[6]from Wall Stree Journal 03.19.87

> "The diversion of funds from the Iran ar ms sales is only a part of the puzzle, and maybe a very small part," a congressional source said. "We first want to focus on how the private network which supplied the Contras got set up in 1984, and whether (President) Reagan authorized it"

If more then one of the rules matches a given part of the document, we randomly pick the first match, which is not necessarily the right choice.

```
Phase: Profile1
Input: Source Token RVG maxNP Quotes Sentence SentStart SentEnd AbbrNAcro
Options: control = appelt

Rule: Profile1
Priority: 100
(START)
((
        (
                (DIRQUOT) |
                (ANY)
        ):cont
        (COMMA)?
        (
                (SOURCE)
        ):source
        (
                (RS_VG)
        ):verb
        (
                (CIRC)?
        ):circ
        (END)
        (
                (START)?
                (
                        (DIRQUOT)
                ):cont2
                (END)?
        )?
):profile)
-->
MAKE_ANNOS
```

Figure 5.2: A JAPE grammar rule to find and mark one type of reported speech sentences

The words that are capitalized are macros for the different purposes. If the transducer discovers such a pattern in the document, it calls MAKE_ANNOS, a macro for a Java function. This function performs the necessary computations to add reported speech annotations to the document.

Keeping in mind that the final goal of the Profile Generator is to obtain statements in the shape of declarative sentences, the Reported Speech Finder is limited to finding reported speech whose reported clause is a complete sentence. This excludes certain reported speech structures whose reported clause is not a grammatically correct statement on its own, like:

$$\text{The President denied to sign the bill.} \tag{5.1}$$

See Section 3.1.3 on Speech Acts for details. Because our system needs grammatically correct sentences to process, we do not consider this kind of reported speech. Alternatively, a transformation algorithm could be used to rephrase the *reported speech act* so a complete reported clause could be gained. This lies outside of the scope of this thesis and could be part of future work 7.3.

5.3.3 Profile Creator

This component is written in Java. It assembles the reported speech annotations and prepares them for the next processing step. That includes extracting single sentences out of the content of the reported speech. It also adds the coreference information for each source. This is done by traversing the data structure build by the fuzzy coreferencer. It adds all coreferring sources as a feature to the reported speech annotations. Another feature that is added by the profile creator is the information whether the reported clause contains a negation or not. The annotation for a reported speech sentence so far can be seen in Figure 5.2.

Type	ReportedSpeech
Start	560
End	654
Source	Doc Schnyder
SourceStart	560
SourceEnd	572
Verb	replied
Cont	In this situation, Preisig has done the right thing.
ContStart	582
ContEnt	636
CorefChainIDs	512
CorefSources	Doc Schnyder, he
Neg	false

Table 5.2: The annotation of a reported speech sentence after running the profile creator

The negation detection function is rather simple. It only looks for keywords within the sentence like “no”, “never”, or “n’t”. This is just a basic way to find negations. More sophisticated approaches are possible, see Chapter 7.3.

5.3.4 Parser Wrapper / PAS Extractor

For the Domain finding process we rely on predicate argument structures extracted from the output of a parser (see Section 5.2). The parser wrapper and PAS extractor component is also implemented in Java. We experimented with three different parsers:

- SUPPLE (more details in Section 5.2.1)
- MiniPar (Section 5.2.2)
- RASP (Section 5.2.3)

The parser wrapper part is used to extract the output of the standalone parser system called RASP. MiniPar and SUPPLE are already well enough integrated into GATE, thus we could use the wrappers that already exist and are shipped with GATE. For the evaluation of our system, we finally decided to use RASP and MiniPar, see Chapter 6. An alternative way would be to use the result of more than one parser, for example together with a voting algorithm (see Chapter 7.3).

The task of the second part of the component is to extract a predicate argument structure from the output of the parsers. This has to be done for all three parsers. To obtain meaningful results we have to rely on the correctness of the parsers' output, which is unfortunately not always the case. In Figure 5.3 the output of RASP, MiniPar, and SUPPLE parser can be seen together with the extracted predicate argument structure.

Sentence	Professor Preisig was a paid consultant to Cilag.
RASP Output	Professor:1 NNS1 Preisig:2 NP1 be+ed:3 VBDZ a:4 AT1 pay+ed:5 VVN consultant:6 NN1 to:7 II-Cilag:8 NP1 .:9 .; (-21.076) ncsubj be+ed:3 VBDZ Preisig:2 NP1 xcomp be+ed:3 VBDZ consultant:6 NN1 det consultant:6 NN1 a:4 AT1 ncmod consultant:6 NN1 to:7 II dobj to:7 II Cilag:8 NP1 passive pay+ed:5 VVN ncsubj pay+ed:5 VVN consultant:6 NN1 obj ncmod consultant:6 NN1 pay+ed:5 VVN ncmod Preisig:2 NP1 Professor:1 NNS1
SUPPLE Output	qlf: name(e2, 'Professor Preisig'), person(e2), realisation(e2, offsets(2, 19)), be(e1), time(e1, past), aspect(e1, simple), voice(e1, active), lobj(e1, e3), consultant(e3), number(e3, sing), adj(e3, paid), det(e3, a), realisation(e3, offsets(24, 41)), realisation(e1, offsets(20, 41)), realisation(e1, offsets(20, 41)), lsubj(e1, e2) qlf: to(e4, e5), ne_tag(e5, offsets(45, 50)), name(e5, 'Cilag'), realisation(e5, offsets(45, 50))
MiniPar Output	s: child_id 696, child_word Professor, head_id 698, head_word was person: child_id 697, child_word Preisig, head_id 696, head_word was pred: child_id 701, child_word consultant, head_id 698, head_word was det: child_id 699, child_word a, head_id 701, head_word was mod: child_id 700, child_word paid, head_id 701, head_word was mod: child_id 702, child_word to, head_id 701, head_word was pcomp-n: child_id 703, child_word Cilag, head_id 702, head_word was
RASP PAS	Preisig – be – consultant
SUPPLE PAS	Professor Preisig – was – consultant
MiniPar PAS	Professor – was – consultant

Table 5.3: Output of three different parser with extracted predicated-argument structures for the sentence shown on top

As can be seen, the parsers have different opinions about the input sentence. This is not an exceptional, or special case, but typical for this task. Notice that we choose a rather simple sentence to demonstrate the different outputs. For more complex sentence structures, the output diversification is even higher and the extracted Predicate Argument Structures look quite different. We will again take a look on this problem in Section 6.3.2.

The PAS extractor uses a set of rules for each of the three parsers by implementing specifically designed subclasses for each parser. These rules determine which part of the parser output is considered the subject, verb, and object. Because of the different nomenclature and relations scheme of the parsers, this has to be done individually for each one.

SUPPLE. For SUPPLE, the extraction process is quite straightforward. The parser outputs *semantic* relations, which comprise a logical subject and verb, and sometimes also an logical object. The PAS extractor therefore only has to filter out these these elements from the output of SUPPLE.

MiniPar. To get predicate-argument structures that represent the underlying sentence as well as possible, we use the following decision tree to select the grammatical structure to use as *object*. If existent and related to the verb of the subject, we choose in this order:

1. "obj";
2. "obj1";
3. "pred"; and
4. "pcomp-n".

Sometimes the *object* does not have a direct relation to the *verb* but an indirect one via a common other element like a "mod" construct. In this case we have to track down and identify this relation to find a representative object. A complex sentence can contain more than one subject and our extractor has to be able to handle them reasonably. And besides dealing with more than one "s" (subject) in one sentence, it can also handle conjunctions. Table 5.4 gives a general overview of the different conjunction types and how the extractor deals with them.

Sentence	**X and Y** buy a car.
Target PAS	X – buy – car Y – buy – car
Sentence	X **buys a car and sells his house**.
Target PAS	X – buy – car X – sell – house
Sentence	X buys **a car and a house**.
Target PAS	X – buy – car X – buy – house

Table 5.4: PAS extractor strategy for conjunctions

RASP. For RASP's version 3 we developed also a wrapper to be able to use it from within GATE. It calls the appropriate script and delivers the parser's output for further processing.

The strategy to find subject, verb, object relations, is to look for "ncsubj" occurrences in the parser output. They describe a subject together with the corresponding verb. To find a suitable object we often have to choose between different elements like "dobj", "iobj", "obj", or "xcomp". To obtain predicate-argument structures that represent the underlying sentence as well as possible, we use the following decision tree on what grammatical structure to use as *object*. If existent and related to the verb of the subject, we choose in this order:

1. "obj";
2. "dobj" if dependent of an "iobj", which itself relates to the relevant *verb*;

3. “iobj”;
4. “dobj”; and last
5. “xcomp”.

Besides dealing with more than one “ncsubj” in one sentence, we can also handle conjunctions. This has already been demonstrated for MiniPar in Table 5.4 and applies for RASP as well.

5.3.5 Fuzzy Believer

The main component of the Fuzzy Believer system is implemented as a couple of Java classes. It is the last component in the pipeline and uses the results of the previous steps to implement different belief strategies. The tasks of this component are:

1. Finding an appropriate Domain for each statement.
2. Computing the fuzzy representation for each statements, identifying their polarity.
3. Process fuzzy information for each Domain according to a strategy.
4. Generate a graphical view of the result.

The following sections provide details of the different tasks and how the implementation solves them. For the design of this component see Section 4.4 and Section 4.5.

5.3.5.1 Domain Finding

The belief processing starts with the finding of domains for the statements (see Section 4.4.1.1) The first step is to name the heuristics that are used for determining the proximity of two statements. By comparing a new statement with existing statements within one Domain, the system can decide whether the new statement is part of this particular Domain or not. The heuristics that are currently implemented can be seen in Figure 5.4 together with the heuristics used by the processing step described in Section 5.3.5.3. The system uses two heuristics for the Domain finding process:

- A WordNet related heuristic, and
- a sub-string heuristic.

To compare subjects and objects of two statements, as defined by the parser output earlier, the sub-string heuristics checks for character overlap between them. In Figure 5.3 the verb WordNet heuristic can be seen.

For the subject, verb, and object of a statement the system uses the WordNet heuristic with different thresholds. These thresholds can be adjusted with run-time parameters. The default configuration for all parameters is shown in Appendix A.

The next step consists of comparing a new statement with statements already assigned to a domain. Another run-time option defines if *strict* matching is necessary to include a new statement in a domain, or if a more *lenient* matching is sufficient. For a strict match, the new statement must match with all existing statements within one domain. In case of a lenient match, the new statement must only match

```
public class Dom_Verb_SynHyp_Heuristic extends Dom_WNHeuristic{
  private double maxSemanticDistance;

  Dom_Verb_SynHyp_Heuristic( HeuristicConfig conf ) throws Exception{
    super( conf );
    maxSemanticDistance =
                myConfig.getDomVerbMaxSemanticDistance().doubleValue();
    // initialize WordNet interface
    try{
      setSemLink( new JWNLEdgeSemanticDistance(
                myConfig.getPropertyFilePath(), myConfig.getWnDictDir()));
    }catch (Exception e){
      e.printStackTrace( Err.getPrintWriter() );
      throw e ;
    }
    // initialize WN cache
    setWnCache( new HashMap() );
  }

  Double corefer( String vg1, String vg2 ) throws ExecutionException{
     double level = 0.0;
     double distance = -1.0;
     try{
       distance = getShortestSemanticDistanceCache( vg1, vg2, PosType.VERB );
     }catch( Exception ex ){
       System.out.println( "Dom_verb_HEURISTIC_SYNHYP: " + ex );
     }
     if( distance >= 0 && distance < maxSemanticDistance+1 ){
           level = (maxSemanticDistance - distance) / maxSemanticDistance;
     }
     return new Double ( level );
  }
}
```

Figure 5.3: Implementation of the domain finding verb WordNet heuristic

with one statement of a domain, implementing a transitive relation on the domain elements.

To obtain a match between two statements, at least two parts of the statements have to be similar enough. That means, the heuristics assigned value must exceed the defined threshold for either subject and object, subject and verb, or verb and object.

The fuzzy set philosophy allows a statement to be part of more than one domain. If a new statement does not fit into an existing domain, a new domain gets founded containing the new statement. Statements and domains are implemented as Java classes and their UML view can be seen in Figure 5.5 and 5.6.

5.3.5.2 Fuzzy Representation

In the next step, the gathered statements for each domain have to be evaluated (see Section 4.4.1.2). The goal is to identify opposing statements by using different heuristics. The disjunction of the positive heuristics provide a degree of similarity between two statements. The fuzzy representation of a statement contains the

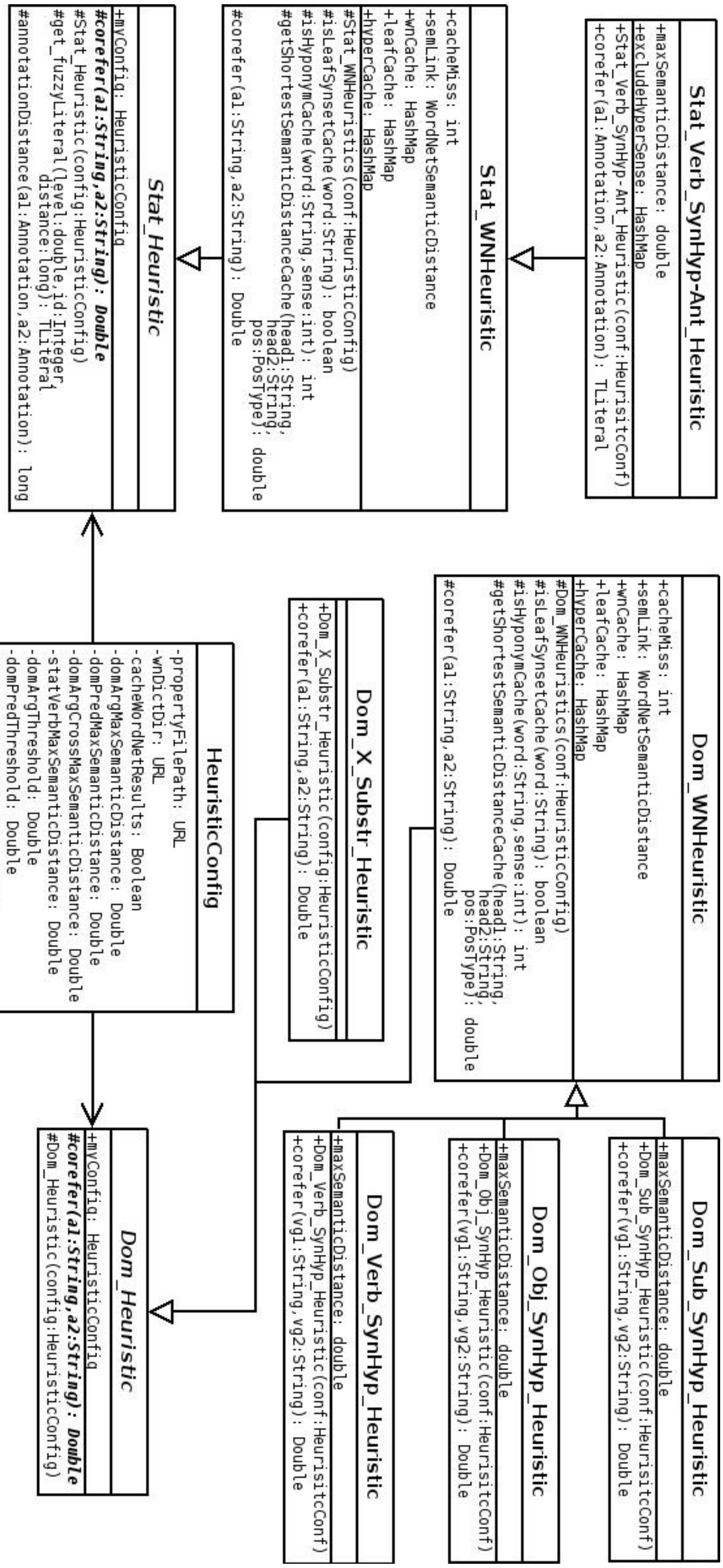


Figure 5.4: Class diagram for the heuristics hierarchy

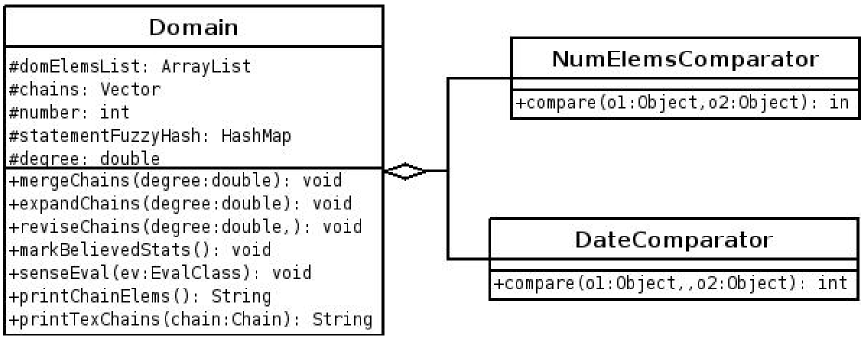


Figure 5.5: Class diagram for the domain class

degrees of similarity of this statement with all other statements within the same domain. In the current implementation, only one positive heuristic is used. It compares the verbs of two statements using the WordNet distance and also incorporates detected negations in the statements. This enables the system to detect negated statements containing synonym words as opposing, for example the two statements:

1. The President didn't intent to insult them.
2. The President wanted to insult them.

The statements are assigned a very low value indicating no similarity in their semantic content. There is also the possibility to use negative heuristics to express no similarity with a high certainty. This is not implemented yet, see Chapter 7.3.

5.3.5.3 Fuzzy Processing

One of the crucial parts of the Fuzzy Believer system is now to decide what out of the collected statements to believe and what to reject (for the design, see Section 4.5) To do so, each domain is processed independently. The domain class implements three different functions according to the three implemented strategies:

1. Believe old news.
2. Believe new news.
3. Believe majority.

The *believe everything* strategy is implemented using fuzzy expansion with a threshold of 0.0, which leads to the inclusion of all statements. *Believe certain newspaper/reporter/source*, and *believe weighted majority* could be implemented using revision together with an ordering of the clauses according to the degree of confidence into the newspaper/reporter/source, but in the current implementation these strategies are not included, see Section 7.3.

For the presented strategies, three fuzzy operations are essential:

- Merging,
- expanding, and
- revising.

These operations are carried out on fuzzy sets, therefore the system works on the generated fuzzy representation. Each statement was assigned a fuzzy representation containing the degree of similarity with all other statements within the domain. Each statement contains a basic belief base consisting only of itself. During the processing steps, these belief bases become combined, processed, or rejected according to a strategy. Its UML class diagram can be seen in Figure 5.6. For the fuzzy set operation, the system makes use of a fuzzy library, described in Witte ([Wit02b]).

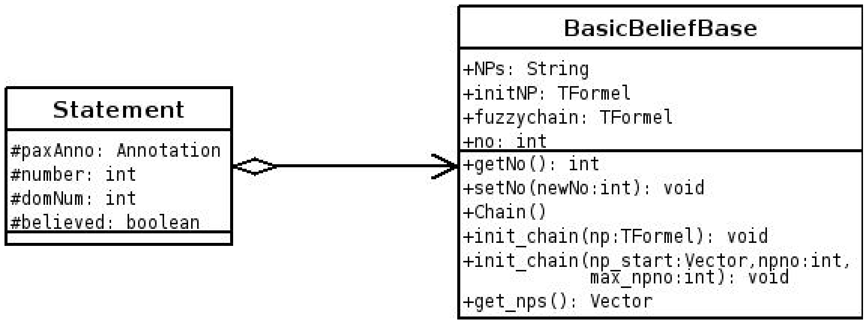


Figure 5.6: Class diagram for the statement class

Believe old news / Expansion. For this strategy we use the fuzzy expansion operation that only adds a new statement to the belief base if it does not contradict the existing statements in a domain. For this processing step we have to ensure, that we process the statements chronologically with the oldest one first. As for all fuzzy operations, a threshold can be adjusted for which a statement is considered similar enough to fit to the existing ones. If a new statement does not reach the threshold the statement is rejected.

Believe new news / Revision. The “believe new news” strategy uses revision to prefer new statements over older ones if they are contradicting each other. Because the ordering of the processing is crucial for the outcome, the oldest statement has to be processed first. If a new statement contradicts the existing statements within its domain, the old contradicting statements are deleted from the belief base and only the ones similar enough to the new one are kept. Again a threshold controls the degree of similarity that is necessary.

Believe majority / Merge. The merging process, implementing the “believe majority” strategy, does not reject or delete statements in the belief base. It rather builds up more than one belief base for each domain containing the different opinions. This is achieved by trying to expand existing belief bases in the domain with a new statement. If the expansion is successful the statement is added to that particular belief base otherwise the next belief base in the domain is checked for similarity. If the expansion of the existing belief bases with the new statement is not successful, a new belief base containing the new statement is added to the domain. After all statements got processed, the algorithm only has to count the statements in each belief base for every domain and marks the statements part of the belief bases with the most elements within one domain as held beliefs.

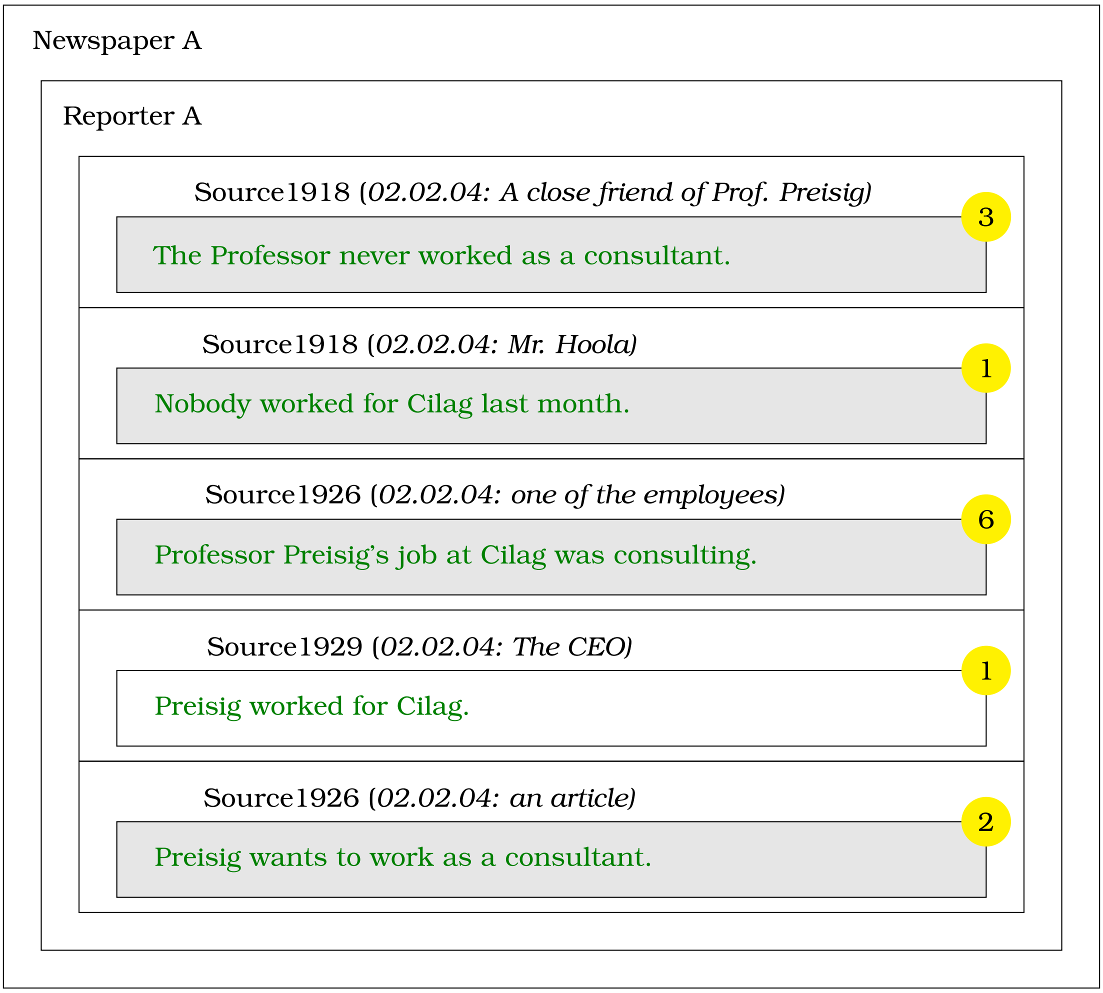


Figure 5.7: Result presentation (the system believes the statements with grey background)

5.3.5.4 Result Presentation

To get an easily readable output of the computed result, the system uses diagrams encoded in LaTeX. A sample system output is shown in Figure 5.7.

The statements that are finally believed by the system are marked by a function that checks the resulting belief base for each domain. This is indicated by a grey background color in the diagram for each statement the system believes. Also the different domains have to be made visible. This is done by assigning numbers to each domain which then can be found in the diagram in the circles on the top right corner of every statement. Also included is the identifying source number together with the date of the article and the actual instance of the source in brackets.

5.4 Summary

We showed in this chapter the component-based implementation of our Fuzzy Believer system. The system is implemented within the GATE framework whose graphical user interface can be seen in Figure 5.8. On the left side, the components are visible, in the middle the document with one Annotation and its features, and on the right side the various annotations of the text. The whole framework is based on Java and to process one medium newspaper article on a standard home PC[7] the system needs around 10 minutes.

[7]Intel Pentium 4, 2GB main memory

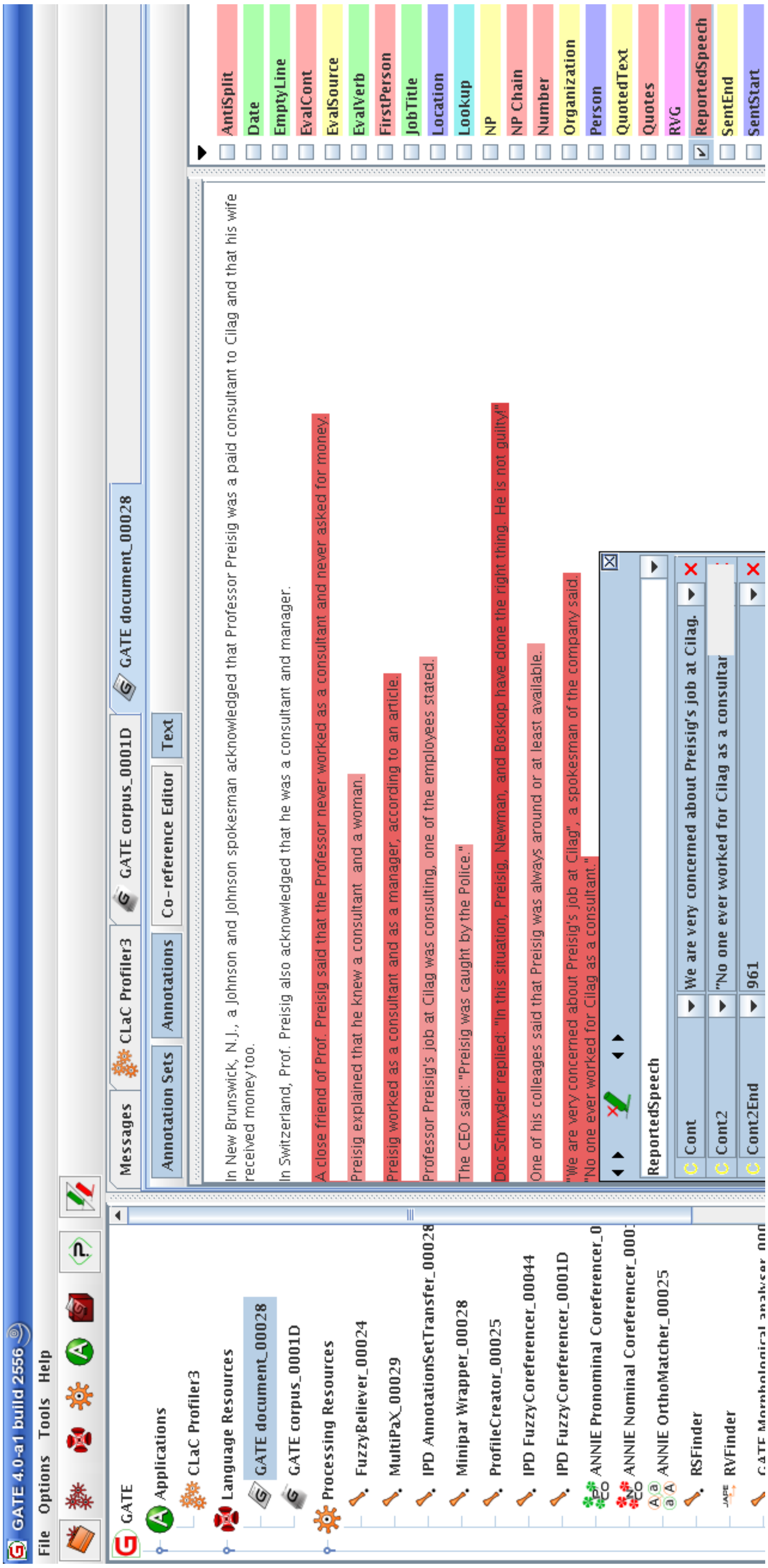


Figure 5.8: Screenshot of the Fuzzy Believer system running in GATE

Chapter 6

Evaluation

This chapter covers the evaluation of ou system, starting with some basic knowledge about important evaluation measures. Then the evaluation process is described in detail: what exactly was evaluated, how, and why. The obtained results and their discussion complete this chapter.

6.1 Background

Basically, we have to deal with two kinds of evaluation scenarios. First, we want to evaluate the performance of extraction components. This is usually done by comparing manually annotated test documents with the annotations the system made for the same document. The overlap of the two annotations yields a performance measure for the system.

The second scenario is not concerned with extracting something out of text, but with correct processing steps. This means the processing can be either right or wrong. To evaluate this kind of task, we need examples that are marked as right or wrong and we need to compare the system's result with this gold standard.

We give an overview of important evaluation measures applied to our system in this section. The advantage of using well established measures is to be able to compare resulting numbers with the results of other systems, and automate the evaluation process by using special tools build for these measures.

6.1.1 Precision

Precision is a measure indicating the stability of the extraction process against marking wrong results as correct. A high value means that not many of those *false positive* results have been found.

Precision is defined as:

$$P = \frac{C}{N} \tag{6.1}$$

where C is a number of properly extracted entities and N is the total number of extracted entities.

For rule-based extraction systems, stricter extraction pattern generally induce higher precision values. Less strict pattern interpretation, on the other hand, induce lower values for precision, since more false positives will occur in the extracted result.

6.1.2 Recall

Recall is a measure indicating how complete the coverage of an extraction process is. That means recall shows how many entities have been found with respect to the total amount of relevant entities present in the text.

Recall is defined as:

$$R = \frac{C}{M} \tag{6.2}$$

where C is a number of properly extracted entities and M is a number of entities described in text.

For rule-based extraction systems, less strict extraction patterns induce higher recall values, since generally more entities match the patterns and are being extracted.

6.1.3 F-measure

F-measure is an integrating measure for both coverage and precision of an extraction system.

F-measure is defined as:

$$F = \frac{(\beta^2 + 1)PR}{\beta^2 P + R} \tag{6.3}$$

where the symbols P and R have the same meaning as above, and β is the relative weight of recall over precision.

Depending on how strict an extraction process is, there is a correlation between precision and recall. Generally, higher precision of a system means lower recall, and vice versa. The F-measure therefore indicates how balanced the extraction process is with respect to coverage and precision.

6.1.4 Accuracy

Accuracy is a measure of how well a binary classification identifies the category of the entities in question, for example if a given sentence is a paraphrase of another sentence or not. In Table 6.1 the different notions involved are summarized.

System Output	**Gold Standard**		
	TRUE	**FALSE**	
Positive	*True Positive*	*False Positive*	Positive Predictive Value
Negative	*False Negative*	*True Negative*	Negative Predictive Value
	Sensitivity	Specificity	Accuracy

Table 6.1: Nomenclature for accuracy measure

Accuracy is computed by following equation:

$$\text{Accuracy} = \frac{\text{\# True Positives} + \text{\# True Negatives}}{\text{\# True Positives} + \text{\# False Positives} + \text{\# True Negatives} + \text{\# False Negatives}} \tag{6.4}$$

6.1.5 Domain Classification

To evaluate the performance of the domain classification algorithm, the above defined precision and recall measures can not be applied. Because a statement can be part of more than one domain, we have to use a different measure. We will use a scoring scheme developed for coreference evaluation. There the problem is comparable to the domain classification. One proposed measure, described in [VBA+95], will also be used for our evaluation task.

Recall. To compute a recall value for the domain classification, we count the minimal number of correct links $c(S)$ necessary to generate the equivalence class of one correct domain. For example, if one domain contains four statements, we need three links to connect these four statements together. $m(S)$ is the number of missing links in the resulting domains. To illustrate this, assume as result three domains, one containing two statements, the other two only one each. If the correct classification would be all four into one domain, two links would be missing. Recall is therefore defined as:

$$R = \frac{c(S) - m(S)}{c(S)} = \frac{(|S| - 1) - (|p(S)| - 1)}{|S| - 1} = \frac{|S| - |p(S)|}{|S| - 1} \tag{6.5}$$

where $|S|$ is the number of elements in one domain, and $|p(S)|$ the number of result domains. For a whole test set T, the corresponding formula is:

$$R_T = \frac{\sum(|S_i| - |p(S_i)|)}{\sum(|S_i| - 1)} \tag{6.6}$$

Precision. For precision, we start by counting the links in the result domain, $c'(S') = (|S'| - 1)$. Then, for the number of missing links $m'(S') = (|p'(S')| - 1)$ we consider the correct domain and how it partitions ($|p'(S')|$), the actual result. Analogous to recall, we have:

$$P = \frac{|S'| - |p'(S')|}{|S'| - 1} \tag{6.7}$$

And for more then one domain in a test set:

$$P_T = \frac{\sum(|S'_i| - |p('S'_i)|)}{\sum(|S'_i| - 1)} \tag{6.8}$$

More complex examples can be found in [VBA+95].

6.2 Corpora

Although it is very easy to obtain a lot of texts through the Internet, it is very hard to get annotated texts that could be used for automatic evaluation. Because of that lack of usable corpora we had on the one hand to find suitable documents and annotate them by ourselves and on the other hand use corpora designed for other purposes to get comparable numbers. For the different subtasks in our system we used three corpora:

- Pascal RTE Challenge Data
- MSR Paraphrase Corpus
- Wall Street Journal Corpus

The first two come with a special annotation of the containing sentences, whereas the last one contains only newspaper articles without annotation of the content. In the following we describe the different corpora in more detailed.

6.2.1 Pascal RTE Challenge Data

The Pascal RTE Challenge Data was designed for a competition between systems to recognize textual entailment. The competition took place in 2005 ([DGM05]) and 2006 ([BHDD+06]), providing us with two data-sets to work with. The task was to decide whether a statement called *hypothesis (H)* can be inferred from a text fragment called *text (T)*. That means if T entails H ($T \rightarrow H$) the system should output "true", otherwise "false". It is crucial to mention that entailment is not a symmetric relation, whereas the relations of interest in our system are symmetric. How this effects the obtained results will be discussed in Section 6.3. The text fragments were chosen and annotated by humans, and entailment was defined in terms of common human understanding of language, including some background knowledge about the world.

In Table 6.2 a few examples are shown from the 2006 corpus, each pair containing an ID, a task category and the entailment result.

Id	T/H	**Text/Hypothesis**	**Task**	**Entailment**
1	T:	The drugs that slow down or halt Alzheimer's disease work best the earlier you administer them.	IR	YES
	H:	Alzheimer's disease is treated using drugs.		
4	T:	Drew Walker, NHS Tayside's public health director, said: "It is important to stress that this is not a confirmed case of rabies."	IR	NO
	H:	A case of rabies was confirmed.		
3	T:	Yoko Ono unveiled a bronze statue of her late husband, John Lennon, to complete the official renaming of England's Liverpool Airport as Liverpool John Lennon Airport.	QA	YES
	H:	Yoko Ono is John Lennon's widow.		
4	T:	Arabic, for example, is used densely across North Africa and from the Eastern Mediterranean to the Philippines, as the key language of the Arab world and the primary vehicle of Islam.	QA	NO
	H:	Arabic is the primary language of the Philippines.		
5	T:	About two weeks before the trial started, I was in Shapiro's office in Century City.	QA	YES
	H:	Shapiro works in Century City.		
6	T:	Meanwhile, in his first interview to a Western print publication since his election as president of Iran earlier this year, Ahmadinejad attacked the "threat" to bring the issue of Iran's nuclear activity to the UN Security Council by the US, France, Britain and Germany.	IE	YES
	H:	Ahmadinejad is a citizen of Iran.		

Table 6.2: Examples for entailment relations

The corpus comprises for each year 800 sentence pairs and is divided into different categories with different levels of difficulty. To classify all pairs correctly, different entailment reasoning is necessary, such as lexical, syntactic, morphological, and logical. The pairs were grouped into different categories, each representing a special application within NLP. For detailed information about the different groups see [DGM05].

6.2.2 MSR Paraphrase Corpus

The fact that the entailment relation is not symmetric was one problem of the PASCAL corpus to use it for evaluation of our system. The resulting numbers might be worse then our actual system's performance (see Section 6.3). This drawback is not present in the MSR Paraphrase Corpus ([DQB04]), because "being a paraphrase of" is a symmetrical relation. The corpus was specifically designed for this aspect and allows evaluation of the our system's domain classification feature (see Section 6.3.2).

The corpus contains 5801 pairs of sentences that were collected during 18 month on various news pages in the Internet. Each sentence pair was manually tagged by multiple annotators judging whether the two sentences can be considered paraphrases of each other or not. Two human annotators were asked for each pair to classify them as either "semantically equivalent" or not. If the two annotators had different opinions about one pair, a third judge was to turn the balance. Out of the 5801 pairs, 3900 were considered semantically equivalent. That is about 67 percent.

<table>
<tr><th>Id</th><th>Sentences</th><th>Equivalent</th></tr>
<tr><td rowspan="2">1</td><td>An autopsy found Hatab's death was caused by "strangulation/asphyxiation," Rawson said $DAY.</td><td rowspan="2">YES</td></tr>
<tr><td>An autopsy found that Nagem Sadoon Hatab's death on $DATE was caused by "strangulation/asphyxiation," Marine spokesman $NUMBER st Lt. Dan Rawson said $DAY.</td></tr>
<tr><td rowspan="2">2</td><td>Mr. Concannon had been doused in petrol, set himself alight and jumped onto a bike to leap eight metres onto a mattress below.</td><td rowspan="2">YES</td></tr>
<tr><td>A SYDNEY man suffered serious burns after setting himself alight before attempting to jump a BMX bike off a toilet block into a pile of mattresses , police said.</td></tr>
<tr><td rowspan="2">3</td><td>A strong geomagnetic storm was expected to hit Earth today with the potential to affect electrical grids and satellite communications.</td><td rowspan="2">YES</td></tr>
<tr><td>A strong geomagnetic storm is expected to hit Earth sometime $DAY and could knock out electrical grids and satellite communications.</td></tr>
<tr><td rowspan="2">4</td><td>But Secretary of State Colin Powell brushed off this possibility $DAY.</td><td rowspan="2">YES</td></tr>
<tr><td>Secretary of State Colin Powell last week ruled out a non-aggression treaty.</td></tr>
<tr><td rowspan="2">5</td><td>Meteorologists predicted the storm would become a category $NUMBER hurricane before landfall.</td><td rowspan="2">YES</td></tr>
<tr><td>It was predicted to become a category 1 hurricane overnight.</td></tr>
</table>

Table 6.3: Examples for true paraphrase relations

The task to decide if one sentence is a paraphrase of another is not clearly definable. Because the sentences are usually at least a little bit different, one might contain slightly different information than the other. The question is whether this information alters the meaning so much that it can not be considered a paraphrase of a given sentence anymore. In Table 6.3 and 6.4 a few examples are shown that should help the annotators to decide about the semantic equality. The examples are from the detailed tagging guidelines in [DBQ05].

In Table 6.3 the example with Id=1 shows that differences in minor details between the sentences are negligible. Or in example two, the main information is equal in both sentences, and some facts are expressed with different words. In the third example, only different lexical items are used to express the same information, therefore the sentences should be considered paraphrases. If one sentence contains anaphora, they can be tagged equivalent, as seen in example four in Table 6.3. The same holds for pronouns, for which the last example in the Table is an example.

1	Prime Minister Junichiro Koizumi has urged *Nakasone* to give up his seat in accordance with the new age rule.	NO
	Prime Minister Junichiro Koizumi did not have to dissolve *parliament* until next summer , when elections for the upper house are also due.	
2	Researchers have identified a genetic pilot light for puberty in both mice and humans.	NO
	The discovery of a gene that appears to be a key regulator of puberty in humans and mice *could lead to new infertility treatments and contraceptives*.	
3	The former wife of rapper Eminem has been electronically tagged after missing two court appearances.	NO
	After missing two court appearances *in a cocaine possession case*, Eminem's ex-wife has been placed under electronic house arrest.	
4	More than *$NUMBER acres* burned and more than $NUMBER homes were destroyed in the *massive* Cedar *Fire*.	NO
	Major fires had burned *$NUMBER acres* by early last night.	
5	A Hunter Valley woman sentenced to $NUMBER years jail for killing her four babies was only a danger to children in her care, a court was told.	NO
	As she stood up yesterday to receive a sentence of $NUMBER years for killing her four babies, Kathleen Folbigg showed no emotion.	

Table 6.4: Examples for false paraphrase relations with crucial terms in italic

Cases that should not be tagged as paraphrases are exemplified in Table 6.4. The first example in that Table shows clearly a different in content, although some entities are present in both sentences ("Koizumi"). Even ambiguous sentences should be tagged not equivalent, holding high the standard for being a paraphrase. In example two, the content is similar, but one of the sentences is a super-set of the other, containing all the information of the second one, but also major details beyond. These cases are not considered paraphrases. The annotators have to decide, whether a detail is a minor detail and can be left out of the consideration, or a detail contains important information and the lack of it in one of the sentences must lead to a “not equivalent” tagging (see example three). Even if there is a possibility that two sentences are referring to the same event, as shown in example four, the likelihood that they express different events is higher and they are not considered paraphrases. The last example in Table 6.4 points out a “not equivalent”-case, where the two sentences are about the same event, but the emphasis lies on different aspects.

More examples and a detailed description of the corpus generation and tagging can be found in [DQB04]. Interestingly, the inter-annotator agreement between the responsible two annotators was around 83 percent, despite the vaguely defined paraphrase relation.

We picked randomly 116 pairs from the corpus and annotated them manually with predicate-argument structures. This was necessary to evaluate the domain finding component which relies on the extracted PAS, without the distortion of automatically extracted PAS.

6.2.3 Wall Street Journal Corpus

The Wall Street Journal (WSJ) Corpus is a collection of newspaper articles taken from the Wall Street Journal between 1986 and 1992. The texts have been chosen, formatted and made available by the Linguistic Data Consortium (LDC), as part of a larger collection with articles from other news sources. The WSJ Corpus, as part of the larger TIPSTER corpus [tip93], was used in projects like TIPSTER[1] itself as well as other NIST[2] sponsored projects like TREC[3]. The WSJ part alone contains about 81 millions words. The format of the single articles contains SGML-like tags to separate the actual content of the document and the meta information like “author”, “date”, ...

To evaluate the reported speech finding component (see Section 6.3.1), we had to annotate articles by hand. In Figure 6.1, the raw XML data of a WSJ article from the corpus is shown, together with the manually annotated reported speech information added by ourselves. We picked randomly 7 newspaper articles (about 6100 words) and marked the reported speech elements:

- Source,
- Reported Verb, and
- Content,

to compare them with our system’s result.

6.3 Results

The evaluation of the Fuzzy Believer system as a whole is quite difficult and complex. Keep in mind that our system should simulate a human newspaper reader and how can one evaluate a human reading newspaper? There is no right or wrong on what to believe – if a newspaper article is the only source for a special information, one is free[4] to believe it or not.

[1]http://www.itl.nist.gov/iaui/894.02/related_projects/tipster/overv.htm
[2]National Institute of Standards and Technology (http://www.nist.gov)
[3]http://trec.nist.gov/
[4]usually background knowledge, context information, world knowledge, and experience influence our decision.

```
<?xml version='1.0' encoding='UTF-8'?>
<GateDocument>
<GateDocumentFeatures>
<Feature>
<Name className="java.lang.String">gate.SourceURL</Name>
<Value className="java.lang.String">file:/home/krestel/Corpora/evalDocs/WSJ870220-0006.xml</Value>
</Feature>
<Feature>
<Name className="java.lang.String">MimeType</Name>
<Value className="java.lang.String">text/xml</Value>
</Feature>
</GateDocumentFeatures>

<TextWithNodes>
<Node id="0"/> President Reagan early last year secretly authorized the Central Intelligence
Agency to kidnap suspected terrorists overseas and bring them to this country to stand trial.

<Node id="181"/>The idea of kidnapping -- described by one law-enforcement official as a "snatch,
grab and deliver operation" -- was approved by the president in a January 1986
directive<Node id="351"/>, <Node id="353"/>according to<Node id="365"/> <Node id="366"/>
administration, law-enforcement and intelligence officials<Node id="424"/>.

<Node id="427"/>The directive, called a "finding," also approved other actions, including covert
operations to preempt terrorist plots, in some cases by attacking the terrorists before they could
strike<Node id="613"/>, <Node id="615"/>the officials<Node id="628"/>
<Node id="629"/>say<Node id="632"/>.

Mr. Reagan approved the finding despite fierce opposition from some officials in his administration
and in the CIA and the Federal Bureau of Investigation.
His decision raised concern among members of the congressional intelligence committees, particularly
over the wisdom of the kidnapping idea and the prospect of preemptive U.S. attacks on terrorists.

<Node id="1000"/>So far, the U.S. hasn't tried to kidnap any suspected terrorists<Node id="1064"/>,
<Node id="1066"/>the officials<Node id="1079"/> <Node id="1080"/>say<Node id="1083"/>.

[...]

</TextWithNodes>
<AnnotationSet>
<Annotation Id="11921" Type="EvalVerb" StartNode="4116" EndNode="4119">
</Annotation>
<Annotation Id="17371" Type="EvalCont" StartNode="4973" EndNode="4983">
<Feature>
<Name className="java.lang.String">repAnnoId</Name>
<Value className="java.lang.Integer">13142</Value>
</Feature>
</Annotation>
<Annotation Id="17485" Type="EvalSource" StartNode="9126" EndNode="9128">
</Annotation>
<Annotation Id="17339" Type="EvalCont" StartNode="11782" EndNode="11845">
<Feature>
<Name className="java.lang.String">repAnnoId</Name>
<Value className="java.lang.Integer">17338</Value>
</Feature>
</Annotation>

[...]
</AnnotationSet>
</GateDocument>
```

Figure 6.1: A Wall Street Journal article (02.20.87) with system generated reported speech annotations in GATE's XML format

One possible scenario to do this evaluation nevertheless would be to find a number of human “reference believers” and present them, as well as the system, controversial newspaper articles. The persons could mark the statements in the articles they believe and we would need a measure to compute the degree of similarity with the system’s output. Our goal is to evaluate the system automatically and not rely on human annotated results. For this task we will concentrate on technical aspects of the individual components and evaluate their performance.

The central tasks we evaluate are

Reported Speech Finding: Implementation described in Section 5.3.1 and in Section 5.3.2;

Domain Finding: Implemented in the Fuzzy Believer component, see Section 5.3.5.1;

Polarity Identification: Also implemented in the Fuzzy Believer, see Section 5.3.5.2;

Fuzzy Strategies: Also implemented in the Fuzzy Believer component, see Section 5.3.5.3.

What might catch the reader’s eye is that the profile generating component is not among the evaluated features. This is due to the fact that the main task of this component is to gather information of the previous steps, namely the Fuzzy Coreferencer results and the reported speech. It processes these information and outputs a well formatted sentence representing the reported speech and information about the source. The performance of the profile generating component relies heavily on the mentioned component running beforehand.

The different evaluation parts are presented in chronological order, starting with the extraction of reported speech from newspaper articles. We conclude the evaluation part with a “real world” example and take a look at the output of our Fuzzy Believer using different strategies.

6.3.1 Reported Speech Finding

Evaluation Strategy. The evaluation for the reported speech finding is straightforward. We compare the system’s output with the gold standard, consisting of newspaper articles manually annotated by us with the reported speech information. To keep it simple, we evaluate only three elements of the reported speech:

- *Source* of a statement,
- *Reporting verb*, and
- *Reported Clause.*

We do not consider *Circumstantial Information* because it is not relevant for further processing of the reported speech in our system. But we do check the *reporting verb*; although it is not used in the current implementation, we might use it in the future (see Section 7.3). The reported speech finding evaluation can therefore be broken down into computing *precision* and *recall* for the reported speech annotation. Apart from correct and incorrect identification of reported speech, we will also introduce partial correctness. If the system annotates a reported speech sentence nearly correct, but, for example, mixes up one or two terms of circumstantial information and reported clause, we speak of partially correct detection, if the meaning of the reported speech in general is maintained.

Results. To evaluate the reported speech finding component, we randomly picked 7 newspaper articles from the Wall Street Journal corpus. The articles contain about 400 sentences (~6100 words) and among them 133 reported speech constructs.

For the detection of reporting verb and source, our system achieved a recall value of 0.83 and a precision value of 0.98. This results in an f-measure of 0.90.

Table 6.5 gives an detailed overview of the results obtained for the different test documents. The results for the extraction of the reported clause (Content) suffers from the misinterpretation of parts of reported clauses as circumstantial information.

WSJ Article	Content			Source/Verb		
	Precision	Recall	F-measure	Precision	Recall	F-measure
861203-0054	1.00	0.50	0.67	1.00	0.63	0.77
861209-0078	1.00	0.77	0.87	1.00	0.79	0.88
861211-0015	0.97	0.88	0.96	1.00	0.89	0.94
870129-0051	1.00	0.71	0.83	1.00	0.71	0.83
870220-0006	0.96	0.74	0.84	1.00	0.93	0.96
870226-0033	1.00	0.58	0.74	1.00	0.58	0.74
870409-0026	1.00	1.00	1.00	1.00	1.00	1.00

Table 6.5: Reported speech extraction results for our system

In Table 6.6 a listing of sample errors that reduce our system's performance is shown.

1. In the first example, the max NP transducer, which the reported speech component uses, failed to identify the NP printed in bold. This leads to a missing match because the patterns of the reported speech finder expect a noun phrase.
2. More complex circumstantial information like in Sample 2 can not be detected by the patterns and the component fails to discover the reported speech.
3. A similar case exists in next sample, where the boundary of the circumstantial information can not be syntactically determined, because after the reporting verb, three occurrences of "that" are possible starting points of the reported clause.
4. This sample contains misguiding quotation marks, excluding the subject of the reported clause "1987". It is not clear, if this is a circumstantial information referring to the date, the utterance was made, or more likely, an addition of the quoted reported clause.
5. The next sample also has complex circumstantial information and also shows the phenomenon of a partly quoted reported clause.
6. Sample 6 was not recognized as reported speech by the system, because the verb grouper component, whose output is used by the reported speech finder, failed to mark the verb construct correctly.
7. The last sample is an example for a rather complex reported speech structure, combining two reported clauses, where the first one does not contain a reported clause that could, grammatically correct, stand on its own. Because of this fact, we do not consider reported speech sentences containing this kind

of “incomplete” reported clauses as valuable reported speech statements and leave this to future work (see Section 7.3). Nevertheless fails our system to detect the second reported clause as reported speech.

No	Example Sentence	Source of Fault
1	“I was doing those things before, but we felt we needed to create a position where I can spend full time accelerating the creation of these (cooperative) arrangements,” **Mr. Sick, 52 years old,** said.	max NP transducer
2	Mr. Furmark**, asked whether he had a role in the arms sale,** said, “I’m not in that business, I’m an oil man.”	reported speech finder
3	However, Charles Redman, the department’s spokesman, said **during a briefing THAT followed the meeting** THAT the administration hadn’t altered its view THAT sanctions aren’t the way to resolve South Africa’s problems.	reported speech finder
4	In his annual forecast, Robert Coen, senior vice president and director of forecasting at the ad agency McCann-Erickson Inc., said **1987 “looks good for the advertising industry.”**	reported speech finder
5	**Praising the economic penalties imposed by Congress last year,** he said **it was “necessary to pursue the question of sanctions further.”**	reported speech finder
6	He **did acknowledge** that he knows Mr. Casey from their past association with Mr. Shaheen.	VG marker
7	He **declined to be more specific, but stressed** that Texas Instruments wasn’t concentrating on alliances with Japanese electronics concerns, but also would explore cooperative agreements with other domestic and European companies.	reported speech finder

Table 6.6: Samples showing different sources of errors for the reported speech finding component

6.3.2 Domain Finding

Evaluation Strategy. How well the domain finding process is working iss evaluated by taking two sentences known to be in one domain. For these sentence pairs we chose the positive examples from a corpus developed for paraphrase evaluation. The results will also show the influence of the different possible parameter settings.

The domain finding process has to rely on the parser output. The different heuristics only use this output, and to give an impression of the dependency, we have hand-annotated some examples with – what we consider – perfect parser results. This comprises the identification of a logical subject, a verb, and one object to form one PAS. We extracted the terms without further changing, that means without giving the root form of a verb, or similar things.

Results. The domain finding task is quite hard and error-prone. Therefore we conducted different test to get an impression of the performance. The domain classification is solely based on the predicate-argument structures extracted from the output of one of the three tested parsers.

The conservative strategy of SUPPLE, only marking relations, which are considers to be 100% correct, proved to be not applicable, because of too few extractable PAS. Table 6.7 shows this problem, as well as pointing out the different results of the two other used parsers. We concentrated therefore on the two remaining parsers for the evaluation.

He said he was paid to advise Cilag on how the company might proceed since "they had done a lousy job" of testing the drug.[a]								
MiniPar			SUPPLE			RASP		
they	done	job	–	–	–	they	do	job
–	paid	Prof.Preisig	–	–	–	Cilag	proceed	drug
company	proceed	–	–	–	–	–	–	–

Table 6.7: A more complex example: PAS extracted from different parsers

[a] from WSJ 12.03.86

The evaluation of the domain finding component includes the comparison of the results obtained with RASP, MiniPar, and manually annotated predicate-argument structures. Table 6.8 shows precision and recall for the two different parsers. The configuration setting means, starting from left to right:

1. Maximum WordNet Distance between subjects.
2. Threshold for merging subjects.
3. Maximum WordNet Distance between verbs.
4. Threshold for merging verbs.
5. Maximum WordNet Distance between subjects.
6. Threshold for merging objects.
7. New statement has to match with one(lenient) or all(strict) statements in one domain.

The test data we used is taken from the MSR corpus and comprised 300 paraphrase pairs. We treated all sentences as content of a reported speech construct. We used the method introduced in Section 6.1.5 to compute comparable recall and precision values. The special layout of the test corpus, containing pairs of paraphrases, made it necessary to develop a method to measure the performance accurately. The fact that one sentence can contain more than one *statement* represented as different predicate argument structures, made the evaluation scenario more complex. The algorithm to compute the precision and recall values is shown in Figure 6.2. It follows the strategy we described in Section 6.1.5.

We also conducted a test with a reduced test set of 116 paraphrase pairs, which was additionally annotated by hand with predicate argument structures by us. This allows an estimation on the influence of the parser and the parser extraction component on the domain classification process. The results can be found in Table 6.9.

```
private void computeDomainTest(){
 HashSet matchSet = new HashSet();
 HashSet pairIdSet = new HashSet();
 int denom = 0;

 Iterator paxAnnosDBIter = this.paxAnnosDB.iterator();
 while( paxAnnosDBIter.hasNext() ){
  Domain myDomain = (Domain)paxAnnosDBIter.next();
  ArrayList domElemsListDB = myDomain.domElemsList;
  int nrDomElems = domElemsListDB.size();
  int nrPairsInDomain = 0;
  HashSet pairMatchSet = new HashSet();

  Iterator domElemsListDBIter = domElemsListDB.iterator();
  while( domElemsListDBIter.hasNext() ){
   Annotation domAnnoDB = ((Statement)domElemsListDBIter.next()).paxAnno;

   String subject = (String)domAnnoDB.getFeatures().get("sub");
   String verb = (String)domAnnoDB.getFeatures().get("verb");
   String object = (String)domAnnoDB.getFeatures().get("obj");
   String cont = (String)domAnnoDB.getFeatures().get("Cont");
   Long pairId = (Long)domAnnoDB.getFeatures().get("PairId");
   pairIdSet.add(pairId.longValue());

   Iterator domElemsListDBIter2 = domElemsListDB.iterator();
   while( domElemsListDBIter2.hasNext() ){
    Annotation domAnnoDB2 = ((Statement)domElemsListDBIter2.next()).paxAnno;
    Long pairId2 = (Long)domAnnoDB2.getFeatures().get("PairId");
    if( (pairId.longValue() == pairId2.longValue()) && !(pairId == pairId2) ){
     matchSet.add( pairId.longValue() );
     pairMatchSet.add( pairId.longValue() );
     nrPairsInDomain++;
     break;
    }
   }
  }
  int nrSinglesInDomain = nrDomElems - nrPairsInDomain;
  int nrPartitions = nrSinglesInDomain + pairMatchSet.size();
  num = num + ( nrDomElems - nrPartitions) ;
  denom = denom + ( nrDomElems - 1 );
 }
 double recall = (double)matchSet.size() / pairIdSet.size();
 double precision = (double)num / denom;
}
```

Figure 6.2: Algorithm used for computing recall and precision values for the domain classification task

Configuration	Results (Recall/Precision)	
	RASP	MiniPar
3-0.5-3-0.5-3-0.5-lenient	0.66/0.34	0.59/0.38
3-0.5-3-0.5-3-0.5-strict	0.62/0.48	0.58/0.52
4-0.5-4-0.5-4-0.5-lenient	0.66/0.34	0.59/0.38
4-0.5-4-0.5-4-0.5-strict	0.62/0.48	0.57/0.52
5-0.4-3-0.4-5-0.4-lenient	0.75/0.24	0.63/0.21
5-0.4-3-0.4-5-0.4-strict	0.62/0.36	0.55/0.40
5-0.5-5-0.5-5-0.5-lenient	0.81/0.38	0.69/0.21
5-0.5-5-0.5-5-0.5-strict	0.61/0.32	0.57/0.38
5-0.5-3-0.5-5-0.5-lenient	0.75/0.24	0.63/0.25
5-0.5-3-0.5-5-0.5-strict	0.61/0.36	0.55/0.40
3-0.3-3-0.3-3-0.3-lenient	0.81/0.37	0.68/0.27
3-0.3-3-0.3-3-0.3-strict	0.62/0.32	0.54/0.36

Table 6.8: Domain classification recall and precision values using RASP and MiniPar

Configuration	Results (Recall/Precision)		
	RASP	MiniPar	Manual
3-0.5-3-0.5-3-0.5-lenient	0.59/0.57	0.54/0.63	0.56/0.78
3-0.5-3-0.5-3-0.5-strict	0.59/0.63	0.50/0.75	0.55/0.85
5-0.5-5-0.5-5-0.5-lenient	0.70/0.29	0.60/0.39	0.62/0.29
5-0.5-5-0.5-5-0.5-strict	0.52/0.41	0.52/0.53	0.52/0.54
5-0.5-3-0.5-5-0.5-lenient	0.65/0.31	0.51/0.57	0.59/0.45
5-0.5-3-0.5-5-0.5-strict	0.59/0.56	0.58/0.41	0.52/0.61

Table 6.9: Domain classification recall and precision values for different parse methods

6.3.3 Polarity Identification

To test the polarity identification or opinion grouping function we need a special corpus containing test data with opposing and supporting statements for a special opinion, but semantically close enough to fulfill the requirements to belong to one domain. The data that comes closest to these conditions are the entailment pairs of the PASCAl challenge corpus. There are some minor drawbacks, though.

First, for the positive entailment examples it is rather easy, to do evaluation, because if one sentence entails another, the senses of the two sentences have the same direction. But non-entailment between two sentences does not necessary infers opposing opinions in these sentences. But fortunately this is often the case for the PASCAL 2 challenge corpus we used. A second problem is the fact that sentence pairs, especially without entailment, would not be put in one domain by our domain classification algorithm and therefore it is not possible to evaluate the data using only the polarity identification component.

We solved these problems by checking the non-entailing examples manually for opposing sentences, and developing a scheme to measure the performance of the polarity identification algorithm without influence from the domain finding component.

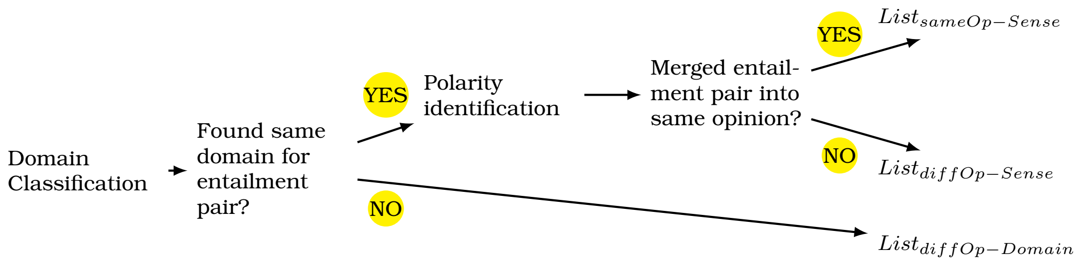


Figure 6.3: Polaritay identification evaluation strategy

Evaluation Strategy. The whole polarity identification process is based on the main verbs of sentences. Other heuristics are possible using different clues, but are not yet implemented. This limits the influence of the parser on the result, because only the verb is used out of the extracted predicate argument structures.

The PASCAL data consists of two kinds of sentence pairs: Entailing and non-entailing. Our Fuzzy Believer system uses a two layer approach to detect similar and opposite statements: Domain finding and polarity identification. One way to evaluate only the polarity identification component is to ignore the results of sentence pairs that were not grouped into the same domain by the domain finding component. This evaluaton algorithm is visualized in Figure 6.3 showing three different lists which we will use for two experiments.

For both experiments, the whole test corpus is processed by the system like a normal newspaper article. Afterwards we count the different entries in the same-opinion-sense list, the different-opinion-sense list, and in the different-opinion-domain list. Together with the correct entailment information from the original corpus, we can compute accuracy of our polarity identification algorithm. The two experiments evaluate different lists from the evaluation algorithm. This ensures that the polarity identification component can be evaluated seperately from the domain finding component. On the other hand, do we get numbers for both components that can be compared to entailment recognizing systems.

Results. We tested different configurations and computed accuracy for two different settings. For one experiment, we included all results in the evaluation counting the entailment pairs that were not put into the same domain by the domain classification as non-entailing. This means we included all three lists in the evaluation process. In the tables this is referred to as “Sense & Domain“. The other test setting only considered the sentence pairs that were actually grouped into the same domain by the domain classification component. That way, we limited the influence of the domain classification algorithm on the polarity identification. We have done that, by only evaluating the results of the two “sense“ lists described in above and shown in Figure 6.3.

An overview of the achieved performance is shown in Table 6.10. In Table 6.11 we give the detailed numbers obtained using the RASP parser and in Table 6.12 the same using MiniPar.

The main difficulty lies in the subtle differences between different opinions expressed in newspapers. Often the different opinions can only be understood by drawing certain conclusions or using inference. Unfortunately only a few examples express clear opposing information like this one from USA Today from 12.05.02:

“President Bush has said Iraq has weapons of mass destruction. (British

Configuration	Accuracy			
	Sense & Domain		Only Sense	
	RASP	MiniPar	RASP	MiniPar
3-0.5-3-0.5-3-0.5-75-strict	0.52	0.55	0.53	0.58
5-0.5-5-0.5-5-0.5-75-lenient	0.51	0.53	0.51	0.53
5-0.5-5-0.5-5-0.5-35-strict	0.52	0.53	0.55	0.51
5-0.5-5-0.5-5-0.5-75-strict	0.51	0.54	0.50	0.56
7-0.5-7-0.5-7-0.5-75-strict	0.51	0.52	0.51	0.52

Table 6.10: Polarity dentification accuracy values for different parse methods

Configuration	Results					
	Positives		Negatives		Negatives (Domain)	
	True	False	True	False	True	False
3-0.5-3-0.5-3-0.5-75-strict	80	70	8	9	142	135
5-0.5-5-0.5-5-0.5-75-lenient	166	147	192	193	18	22
5-0.5-5-0.5-5-0.5-35-strict	84	73	11	5	137	139
5-0.5-5-0.5-5-0.5-75-strict	84	73	124	139	68	58
7-0.5-7-0.5-7-0.5-75-strict	87	70	183	193	26	19

Table 6.11: Detailed polartity identification results obtained with RASP parser

> Prime Minister) Tony Blair has said Iraq has weapons of mass destruction. (Secretary of Defense) Donald Rumsfeld has said Iraq has weapons of mass destruction," White House spokesman Ari Fleischer said Thursday. "Iraq says they don't. You can choose who you want to believe."

Often the different sense is very hard to grasp without semantic understanding. Here one example from the pascal corpus (*pair id="505" entailment="NO" task="IE"*):

> "The European Commission, on Tuesday, warned Bulgaria and Romania that their accession to the EU could be delayed until 2008 if they fail to take urgent action to fight corruption and speed up reforms." "The European commission blocks the Romanian accession to the EU."

6.3.4 Fuzzy Strategies

Evaluation Strategy. To visualize the different strategies, we choose different controversial newspaper articles dealing with the same topic. The different output of the system should give an impression of the various strategies.

Configuration	Results					
	Positives		Negatives		Negatives (Domain)	
	True	False	True	False	True	False
3-0.5-3-0.5-3-0.5-75-strict	72	45	10	15	165	142
5-0.5-5-0.5-5-0.5-75-lenien	135	113	122	119	64	56
5-0.5-5-0.5-5-0.5-35-strict	67	45	117	134	82	62
5-0.5-5-0.5-5-0.5-75-strict	69	46	17	23	161	139
7-0.5-7-0.5-7-0.5-75-strict	68	51	109	114	87	75

Table 6.12: Detailed polaritiy identification results obtained with MiniPar parser

The following section presents the results our system obtained. We used the Wall Street Journal corpora described above to conduct various evaluations.

Results. There is no measure yet to evaluate the different strategies, so we decided to present a real world example together with the results of the system's different strategies. We picked 17 newspaper articles from the Wall Street Journal between 01.07.87 and 05.04.87 related to the Iran-Contra Affair. Remember the example from Chapter 1, were we picked excerpts from one of the Iran-Contra Affair related articles from the Wall Street Journal. Presenting the whole results for every strategy would be beyond the scope of this Section, therefore we limit the result presentation to a couple of sentences from these articles. We show for each statement the results of the different strategies, whether the system believes the given statement or not. These results are not only dependent on the chosen strategy, but also heavily dependent on the domain classification and polarity identification results, as well as on the reported speech extraction and the quality of the extracted predicate argument structures. The results can be found in Table 6.14. The tested strategies comprise:

- Old News (referred to as "Old");
- New News (referred to as "New"); and
- Majority (referred to as "Maj").

An "X" in a field of the Table indicates that the particular strategy believes the given statement.

In Table 6.13 we show one system-generated domain. The parameters, for the sake of completeness, were 5.0 for the maximum WordNet distance for all elements, and 0.5 for the thresholds for all elements.

Something we have to be aware of is the fact that one sentence can contain more than one statement represented as PAS. For example the sentence:

> It may be difficult to prove the president didn't know about and didn't authorize a diversion.

contains the aspect of knowing and the aspect of authorizing. We have therefore two predicate argument structures for this sentence and this usually leads to two different domains, this sentence is part of. The fuzzy processing is responsible that even one PAS can be part of different domains. These two facts might lead to opposing results concerning the belief status of a sentence by the system. One sentence as part of one domain might be believed by the system, whereas the same sentence as part of another domain might have been rejected by the system. For our evaluation we choose to mark all sentences as "not-believed" as soon as they are not believed as part of one domain. The Table 6.14 contrasts the result of the different strategies. The table shows an excerpt from the results of the Wall Street Journal articles about the Iran-Contra Affair that were processed by the system. The original newspaper articles excerpts together with the original system output samples can be seen in Appendix E.

6.4 Discussion

The results our system achieved for extracting reported speech is highly competitive. Doandes [Doa03] reports a recall of 0.44 and a precision of 0.92 for their system compared to 0.83 and 0.98 our system obtained.

Domain Nr. 10
President Reagan was deeply involved in the unfolding policy toward Iran.
The committee's narrative, gave an opinion last January that withholding congressional notification of the arms sales was legal.
Senior CIA officials were worried about the private individuals assisting Col. North in the Iran and Contra operations.
Vice chairman made the decisions, and vice chairman wasn't manipulated.
Vice chairman made the fundamental decision that the Contras shouldn't perish.
They don't yet have any evidence that Mr. Reagan knew profits from the sale of arms to Iran were diverted to help the Contras, or that Mr. Reagan approved any law-breaking by his aides or others.
McFarlane told the president everything he knew.
The new hearings, however, will show that the president was actively involved in the policies, and not a "befuddled" bystander.
It may be difficult to prove the president didn't know about and didn't authorize a diversion.
Current and former officials, told Col. North several times in 1985 that Col. North couldn't become a broker for aid to the Contras or get into the business of actually buying and selling weapons.
President didn't believe such donations were illegal because they were private.
President Reagan knew "that there were many people privately giving money" to aid the Contras during the period Congress had cut off U.S. funds to them.
Mr. Reagan knew nothing of any diversion before Justice Department officials found evidence of it in Col. North's files last November.
Israel knew anything about the diversion.
Mr. Reagan had "no detailed information" that the money was used for weapons.
It isn't clear that Mr. Reagan knew whether the administration was participating in that effort.
The Sandinistas use the civil war against the Contras as an excuse to avoid negotiations on relaxing domestic political restrictions.
The donation was made for medical supplies.

Table 6.13: One domain within the Iran-Contra Affair articles

For the domain classification, our best results range from a recall of 58% to 81% and a precision of 38% to 52%. These values can probably be improved by using more sophisticated heuristics, although there will be a ceiling set by the parser and by the use of language in general. The same meaning can be expressed by various different sentences whose words are not in close relations to each other and therefore hard to detect by current NLP tools. Keeping these facts in mind, the obtained numbers are rather satisfactory and promising for future development.

The rather shallow semantic approach sets a practical limit to the achievable results. This can be infered by comparing the numbers obtained using manually parsed predicate-argument structures with the numbers obtained by the parsers. It shows that there is space for improvement on the side of the parsers, as well as on the side of the PAS extractor. But a precision of 55% and a recall of 85%, as obtained for the best configuration of the system using manually parsed PAS, shows that it needs more and/or better heuristics to get a really significant improvement.

The polarity identification task was expectedly the hardest one. This can be seen at the rather poor results we obtained by trying to find different opinions

Statement	**Strategies**		
	Old	**New**	**Maj**
President Reagan was deeply involved in the unfolding policy toward Iran.	-	-	X
Mr. Reagan was briefed about the arms-hostages link.	X	X	X
Both Mr. Regan and the vice president knew nothing of the diversion.	X	X	-
It may be difficult to prove the president didn't know about and didn't authorize a diversion.	-	-	X
Mr. Reagan had "no detailed information" that the money was used for weapons.	-	X	X
The committee's narrative, gave an opinion last January that withholding congressional notification of the arms sales was legal.	-	-	X
Mr. Reagan's personal diary, his appointment calendar and his public statements all show Mr. Reagan's was actively involved in raising funds for the Contras and in encouraging his aides to do likewise.	X	X	-
Funds diverted from the Iranian arms sales and other moneys were routed through Swiss bank accounts and a network of companies in the U.S., Europe and Central America.	-	-	-
Adm. Poindexter kept some sensitive documents, including the only copy of a January 1986 presidential directive authorizing U.S. arms sales to Iran, in a safe in his office.	-	X	X
Mr. Meese never provided any legal advice or violated any congressional bans related to Col. North's efforts to help ship weapons and other supplies to the Contras.	X	X	X
Mr. Reagan had "no detailed information" that the money was used for weapons.	X	X	-
Laws may have been breached, though the commission carefully avoided reaching conclusions on legal issues.	X	-	-
The Contras are a corrupt fighting force whose leaders have as little interest in bringing democracy to Nicaragua as the Sandinistas appear to.	-	X	X
The president's letter and the reference to the vice president's meeting were suppressed for "diplomatic" rather than political reasons.	-	-	X
While Gen. Secord and associates assembled a complex network of companies, bank accounts, and individuals to funnel weapons and supplies to the rebels, Mr. Casey personally recruited the CIA station chief in Costa Rica to assist.	-	-	X
The donation was made for medical supplies.	-	X	X
Administration officials were generally aware of the private supply operation, but didn't know many details about it.	-	-	-
Current and former officials, told Col. North several times in 1985 that Col. North couldn't become a broker for aid to the Contras or get into the business of actually buying and selling weapons.	X	-	-

Table 6.14: System result for WSJ articles about the Iran-Contra Affair

within one domain. Best accuracy values were obtained using Minipar and were around 58%. This task is very hard for computational systems. But with more elaborated heuristics it is possible to increase these numbers, comparable to the Pascal challenge [DGM05, BHDD+06], where systems also started with around 50% accuracy and became better over time.

The testing of the different strategies revealed that the fuzzy processing operators are working as expected. Further evaluation of the results would need some kind of measure to get quantitative, compareable results. This is beyond the scope of this thesis and defered to future work (see Section 7.3).

Chapter 7

Conclusions and Future Work

This chapter proposes general analysis of the work and outlines directions of further work.

7.1 Review of the work

In this section we present an objective review of the work done within the project. The goal of this thesis was to get a working system that uses the structure of reported speech in newspaper articles to build a belief database. To achieve this, fuzzy theories was used to model the inherent vagueness of natural language and the lack of real language understanding by computational systems.

To go back to where we started, remember the newspaper article excerpt in Figure 1.1. After our Fuzzy Believer system has analyzed the article together with other articles[1] about the Iran-Contra Affair, the system holds certain beliefs and rejects other statements of the article. In Figure 7.1 and 7.2, the output of the Fuzzy Believer is shown applying the "believe new news" strategy. Only the statements of the example article from Chapter 1 are considered for this figure.

The computation of the beliefs was achieve by using a component-based system implemented within the GATE framework. We developed five components in accordance to the subtasks that had to be solved:

- Reported Verb Vinder,
- Reported Speech Finder,
- Profile Creator,
- Parser Wrapper / PAS Extractor, and
- Fuzzy Believer.

These components extract reported speech out of newspaper articles and analyze and process them using fuzzy set theory. For this purpose, we also incorporated external parsers to extract predicate-argument structures from the reported statements. We then performed the computation based on them deploying fuzzy operations. We implemented different *belief strategies* to model different types of human newspaper readers. We also evaluated our system extensively using different methods.

[1] can bee found in Appendix E

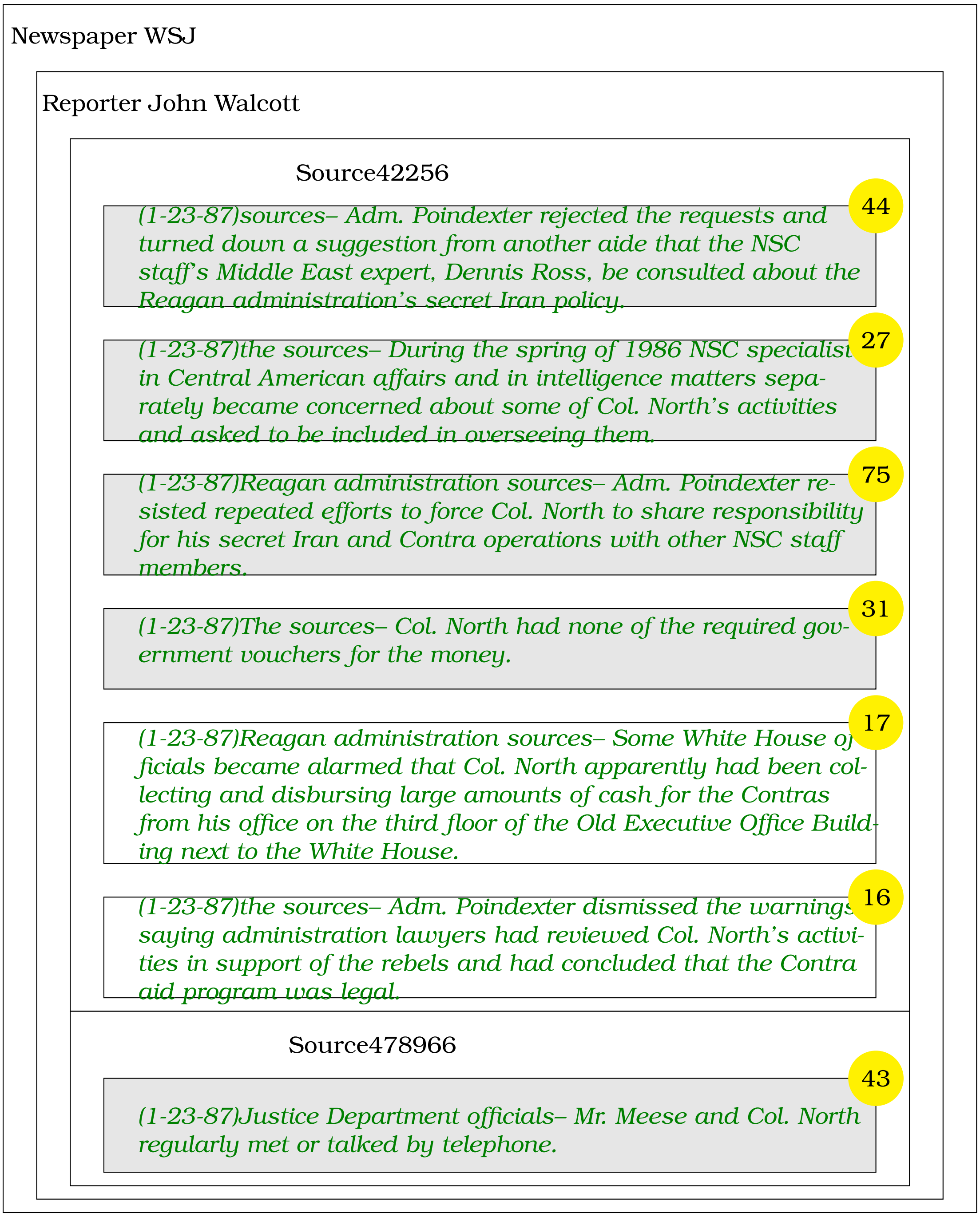


Figure 7.1: Believe new news strategy: Result for example article (a)

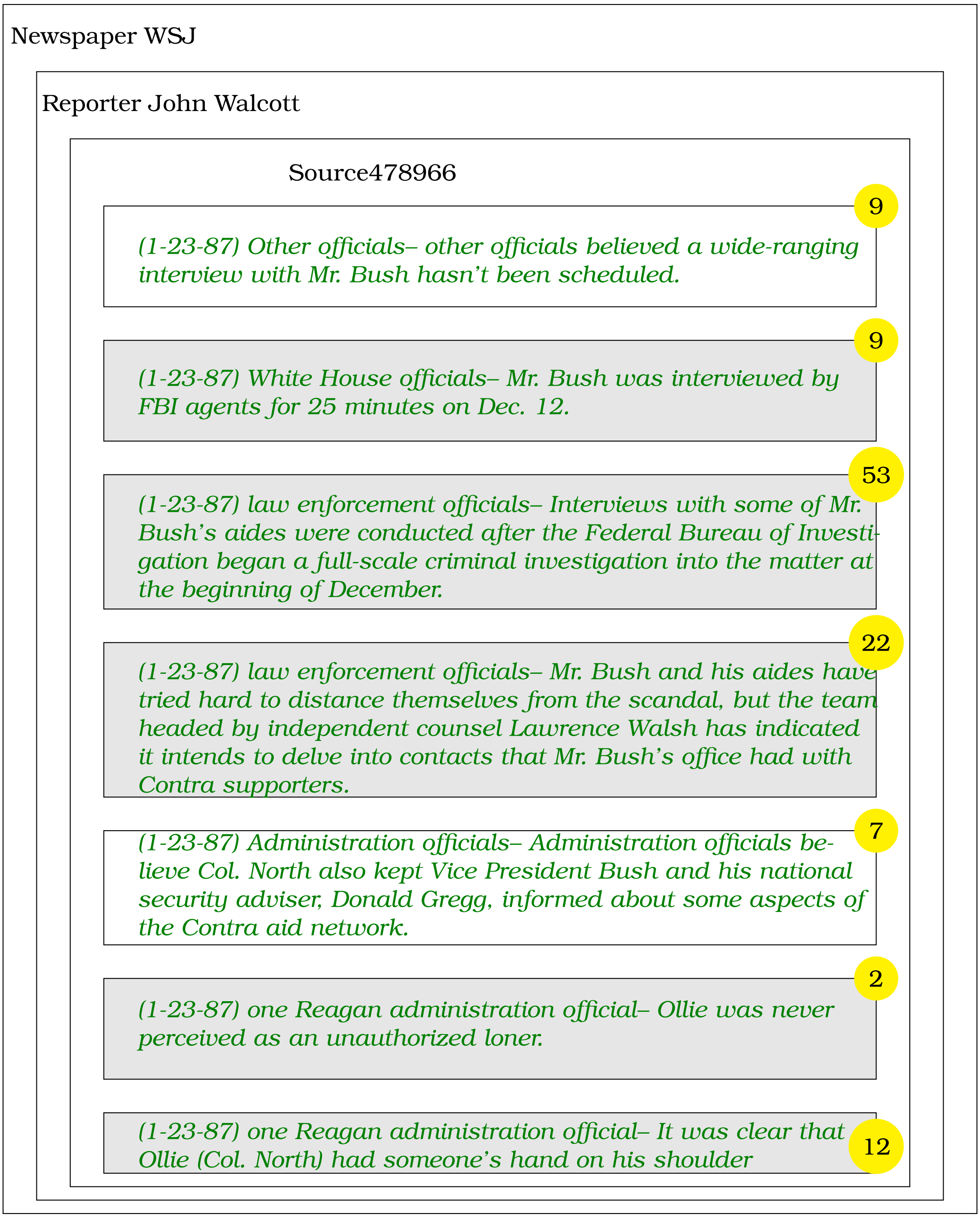


Figure 7.2: Believe new news strategy: Result for example article (b)

7.2 Overview of results

We achieved manifold results – from the evaluation of the individual components as well as from the complete system. The results for each component are discussed below.

Reported Speech Extraction. We implemented rules to identify and *extract reported speech* from newspaper articles using JAPE. Therefore we distinguished between 6 basic patterns. Apart from extracting the reported clause, we also get the source, the reporting verb, and circumstantial information. For our test corpus we achieve 83% recall and 98% precision for the identification of source and reported clause. These values are better than the reported recall and precision values from Doandes [Doa03] (recall: 44%; precision: 92%) who also extracted reported speech automatically.

Domain Classification. We designed and implemented various heuristics to compute the relatedness between two predicate argument structures. We used fuzzy set theory to *group related PASs into one Domain*. The result should be one Domain for each individual topic or opinion within a newspaper article. Some work has to be done beforehand to get predicate argument structures out of the extracted reported clauses. For this task we implemented a PAS extractor which uses the output of one of the parsers RASP, SUPPLE, or MiniPar to generate a PAS structure. This component achieved within our test scenario results ranging from a recall of 58% to 81% and a precision of 38% to 52%. In [**?**] they present a paraphrase detection system achieving similar results, but because of a different task of the system and different evaluation measures we can not compare numbers.

Polarity Identification. The next processing step required the *determination of the sense* expressed in each statement within one Domain. Again we used heuristics, this time to identify opposing or supporting senses or opinions between the statements. We used different parameters to vary the threshold, exploiting the fuzziness modeled in our approach. Here the numbers are not as impressive as for the previous steps: The best result was obtained using MiniPar, yielding an accuracy of 58%. The PASCAL Recognising Textual Entailment Challenge [DGM05, BHDD$^+$06] deals with similar questions. For entailment recognition, the participating systems achieved in 2005 accuracy values between 50-59% and in 2006 52-75%. With only one heuristic used for polarity identification, our results are highly competitive.

Belief Strategies. For each Domain fuzzy operations allowed the system to decide which statement to believe or to reject, according to different *belief strategies*. These strategies were implemented using standard fuzzy set procedures to model different types of artificial believers. The strategies implemented are “believe in the majority”, “believe new news and reject old opposing news”, and “prefer old news over new news”. The evaluation for the strategies showed that it is useful to model different strategies to reflect different attitudes of humans towards belief. There is no similar system to compare our final results with, but current research activities[2] may offers the possibility of comparing our system with others in the near future.

[2] http://www.nytimes.com/2006/10/04/us/04monitor.html

7.3 Future work

This section presents directions of future work, which would advance system functionality and improve system performance.

Each single component in the system can be improved, because within NLP a component scarcely is working 100 percent correct. The richness of human language, with all it's ambiguities, makes this for most processing components an unreachable task. But there still is room to improve each single component by adding more rules or getting better statistical data, better lexicons, or by covering more special cases. In this section, we are not concerned with those types of improvement that could be gained by tweaking the system here and there. But we are rather interested in additional features and improvement of concepts.

Improvement of the Fuzzy Believer System. Interesting aspects for future work to improve our system are presented in this paragraph with respect to our system's architecture, from the lower levels of processing to the higher levels.

A new direction for future development would be to extend the system to cover statements not in the scope of reported speech, increasing the base of possible beliefs. But also each single component could be improved:

- The reported speech detection patterns could be extended to cover more special cases of reported speech structures.
- More heuristics to identify the sense of a statement, like taking into account temporal aspects, more than one argument for a predicate if available, or – more sophisticated – using world knowledge and logical inference to detect the relation between two statements could improve the system's performance significantly.
- Using the results of more than one parser simultaneously is an idea to get more robust results for the domain finding process.

Another approach could use a threshold for a statement to be believed, introducing "potential" and "hold" beliefs. So some evidence must be provided before an utterance actually becomes a belief for the Fuzzy Believer.

Additionally, using evidential analysis together with majority strategy. assigning values to each statement according to newspaper agency, reporter, source, and reported verb. Implementing this *weighted majority* strategy, as we called it in Chapter 4.5.1, would lead to a more "personalized" Fuzzy Believer. The result would be different according to the "experience" the Fuzzy Believer had in the past. The credibility value assigned to each involved entity could get adopted, mirroring obtained conclusions of the Fuzzy Believer. For example, a highly trusted source could get a lower credibility value assigned, if it utters, with respect to the reported verb, statements not compliant with the majority.

Application of the Fuzzy Believer. What is still missing, is a way to evaluate our final results. Finding a method for the evaluation of general belief processing systems is an important task for the future. This probably includes manually annotated corpora and some measure to compare results of different systems.

The Internet as a field of application is very interesting for our Fuzzy Believer system. One scenario could be to use agents who interact with each other and exchange their beliefs while crawling through the Internet.

The global media community also offers lots of application scenarios. The Fuzzy Believer could analyze news from different regions and compare them with each other, e.g., Occident and Orient.

Companies could also use the system to analyze opinions of their products or the products of their competitors. Therefore not only forums and homepages could be exploited but also blogs or customer's e-mails.

In conclusion, the developed Fuzzy Believer system represents a promising new approach to deal with the growing information load and the increasing importance of information for administrations, governments, companies and individuals.

Appendix A

Fuzzy Believer Component Parameters

Run-time parameter accessible via GATE of the Fuzzy Believer Component together with the default values are shown in Figure A.1.

Parameter Name	Type	Default	Comment
inputASName	String	-	The input annontation set containing the 'sentence' annotation.
outputASName	String	-	The output annotation set for this component.
jwnlProperty-FilePath	URL	-	Location of the jwnl property file.
wnDictDir	URL	-	Location of the WordNet 2.0 dictionary database diretory.
cacheWordNetResults	Boolean	false	Locally cache the WN results in the Fuzzy-Coreferencer?
debugFlag	Boolean	false	Show debug information.
strictMatch	Boolean	true	Strict domain creation; candidate has to match all domain elements not only one.
dom_Sub_Heuristic-SynHyp_MaxDist	Double	5.0	Max semantic distance for Sub WordNet heuristic.
dom_Verb_Heuristic-SynHyp_MaxDist	Double	3.0	Max semantic distance for Verb WordNet heuristic.
dom_Obj_Heuristic-SynHyp_MaxDist	Double	5.0	Max semantic distance for Obj WordNet heuristic.
stat_Verb_Heuristic-SynHyp_MaxDist	Double	3.0	Max semantic distance for Verb WordNet heuristic.
stat_Merge-Revise-Expand-Degree	Double	0.6	Threshold for processing ChainS.
subThreshold	Double	0.5	Threshold for subject domain membership.
verbThreshold	Double	0.5	Threshold for verb domain membership.
objThreshold	Double	0.5	Threshold for object domain membership.
pasElemsToMatch	Integer	2	How many element pairs (1-3) need to have a value higher than the threshold to become one domain.
believerTypeList	ArrayList	majority	The model of the artificial believer. Possible values are: 'majority', 'oldNews', or 'newNews'.

Table A.1: Default parameter setting for the fuzzy believer component

Appendix B

Fuzzy Believer Application

A detailed overview of the processing resources used for the Fuzzy Believer Pipeline in GATE can be found in Table B.1.

Table B.1: Processing resources of the application pipeline

Processing function	GATE resource name	Control input
Document resetting	CLaC Document Resetter	—
Tokenization	ANNIE English Tokenizer	Standard tokenization rules for English texts
Number combining	JAPE Transducer	Grammar for identifying numbers
Number interpreting	CLaC Number Interpreter	rules for the interpretation of numbers
Hyphenated-tokens combining	JAPE Transducer	Grammar to combine hyphenated tokens
Abbreviation and acronym marking	CLaC AAMarker	List of abbreviations and acronyms
Gazetteering	ANNIE Gazetteer	Gazetteer list for locations, dates and so on
Named Entity recognition 1	ANNIE NE Transducer	rules for identifying named entities part 1
Sentence splitting	CLaC Sentence Splitter	Standard splitting rules for English sentences
Part of speech tagging	ANNIE POS Tagger	Brill tagger trained on general English texts
Named Entity recognition 2	ANNIE NE Transducer	rules for identifying named entities part 2
Sentence split checking	CLaC Sentence Split Checker	rules to ensure a correct sentence split
Verb Phrase chunking	ANNIE VP Chunker	Grammer to identify Verb Phrases
Noun Phrase chunking	JAPE Transducer	Grammar to identify Noun Phrases
Maximum Noun Phrase transducing	JAPE Transducer	Grammar to find max NP
Morphological analysing	GATE Morphological Analyser	Rules to add the lemma for each token

Continued on next page ...

Continuation of Table B.1 ...

Processing function	**GATE resource name**	**Control input**
Finding reported verbs	JAPE Transducer	Grammar to find verbs used in reported speech
Finding reported speech	Montreal Transducer	Grammar to find reported speech
Named Entity coreferencing	ANNIE Orthographic Matcher	—
Nominal coreferencing	CLaC Nominal Coreferencer	—
Pronominal coreferencing	CLaC Pronominal Coreferencer	—
Fuzzy coreferencing	CLaC Fuzzy Coreferencer	fuzzy parameters
Profile creation	Profile Creator	—
Parser wrapping	Rasp3 Wrapper	—
Fuzzy believe computation	Fuzzy Believer	strategies

Appendix C

Components used for preprocessing in the fuzzy believer system

C.1 Tokenizer

The tokenizer annotates tokens of a text according to their symbolic structure. Therefore, it creates the "Token" Annotations with the Features "orth" and "kind". It is kept simple, so that it is on the one side flexible enough for all kinds of different tasks and on the other side very efficient. It leaves the more complex work to the JAPE Transducers (see Section 5.1.1.2).

The "orth" feature draws a distinction between words in lower-case characters and upper-case characters. Possible "kind" feature values are number, word, punctuation, space-token, or symbol.

The current English tokenizer used is actually the ANNIE English tokenizer shipped with GATE, which runs on a slightly modified grammar, not doing any concatenation if an dash "-" is found between two words.

The rules of the tokenizer have a left hand side (LHS) and a right hand side (RHS). The LHS is a regular expression which has to be matched on the input; the RHS describes the annotations to be added to the annotation set. The LHS is separated from the RHS by ">". The operators that can be used on the LHS are the same operators as in JAPE grammars (see Section 5.1.1.2. The RHS uses ";" as a separator, and has the following format:

LHS > Annotation type; attribute1 = value1; ...; attribute n = value n

The following tokenizer rule is for a word beginning with a single capital letter:

*"UPPERCASE_LETTER" "LOWERCASE_LETTER" * > Token; orth=upperInitial; kind=word;*

It states that the sequence must begin with an uppercase letter, followed by zero or more lowercase letterS. This sequence will then be annotated as type "Token". The attribute "orth" (orthography) has the value "upperInitial"; the attribute "kind" has the value "word".

In the default set of rules, the following kinds of Token and SpaceToken are possible:

Word A word is defined as any set of contiguous upper or lowercase letters, including a hyphen (but no other forms of punctuation). A word also has the attribute orth, for which four values are defined:

- upperInitial – initial letter is uppercase, rest are lowercase
- allCaps – all uppercase letters
- lowerCase – all lowercase letters

- mixedCaps – any mixture of upper and lowercase letters not included in the above categories

Number A number is defined as any combination of consecutive digitS. There are no subdivisions of numberS.

Symbol Two types of symbols are defined: currency symbols (e.g., "$", "£") and other symbols (e.g., "&", "§").

Punctuation Three types of punctuation are defined: start punctuation (e.g. "("), end punctuation (e.g. ")"), and other punctuation (e.g. ":"). Each punctuation symbol is a separate token.

SpaceToken White spaces are divided into two types of SpaceToken – space and control – according to whether they are pure space characters or control characterS. Any contiguous (and homogeneous) set of space or control characters is defined as a SpaceToken.

The above description applies to the default tokenizer. However, alternative tokenizers can be created if necessary. The choice of tokenizer is then determined at the time of text processing.

This tokenizer is used together with a special English JAPE transducer (see 5.1.1.2) as a single processing resource called English tokenizer. The transducer has the role of adapting the generic output of the tokenizer to the requirements of the English part-of-speech tagger. One such adaptation is joining together constructs like "30's", "'Cause", "'em", "'N", "'S", "'s", "'T", "'d", "'ll", "'m", "'re", "'til", "'ve", etc. in one token. Another task of the JAPE transducer is to convert negative constructs like "don't" from three tokens ("don", " ' " and "t") into two tokens ("do" and "n't"). The English tokenizer should always be used on English texts that need to be processed afterwards by the POS Tagger.

C.2 Gazetteer

This component is used for tagging tokens with their semantic categories. It creates the "Lookup" Annotation over certain tokens with the "majorType" and "minorType" features. Values of these Features define major and minor semantic categories of the token. The resource utilizes a set of list files: each file containing a set of names that have a certain type; and the definition file (lists.def), which attaches the "majorType" and "minorType" values to each names file.

The ANNIE Gazetteer simply performs the keyword look up among the document. It does not take into consideration any contextual, lexical, or semantic information, just brutal exact match.

Our gazetteer is just an ANNIE Gazetteer running on lists that have been enriched. Moreover, some categories have been refined, such as breaking the general type "AM_PM" down to two atomic types "AM" and "PM", for better usage.

C.3 Named Entity Transducer

Named entities are textual phrases, which refer to persistent concepts of the world. Named entities have a specific semantic meaning and correlate to concepts of an information extraction domain. Named entity recognition is a separate task within information extraction. It includes locating named entity phrases within text and classifying them into predefined categorieS. Named entity recognition systems have been created based on linguistic rule-based techniques as well as statistical models.

Gazetteering is often used as a preliminary procedure to named entity recognition. The rule-based approach to named entity recognition consists in using grammars with rules defining the sequences and atoms, which include results from gazetteering.

It is convenient to use annotations, named with entity category names, to denote named entities in a text.

The processing units are ordered in the following way:

- NE recognition without unknown word capacity
- sentence splitting
- POS tagger
- NE recognition with unknown word capacity (now the POS information is available)
- sentence split checker

The results of the NE recognition and the sentence splitter influence each other. That's why we have to verify the sentence split results after the NE recognition. For example a name like "Dr. F. Meyers" should get detected by the NE recognition and the full stops should then get recognized by the sentence splitter as not marking a sentence boundary.

C.4 Sentence Splitter

The resource is a cascade of finite-state transducers, which segments text into sentenceS. Eventually, it creates the Annotation "Sentence" attached to the sentence boundaries, which is used by other Processing Resources.

The splitter uses the abbreviation/acronym marker 5.1.2 annotations to distinguish sentence-marking full stops from other kindS. Each sentence is annotated with the type `Sentence`. Each sentence break (such as a full stop) is also given a Split annotation. This has several possible types: ".", "punctuation", "CR" (a line break) or "multi" (a series of punctuation marks such as "?!?!". The sentence splitter is domain- and application-independent.

C.5 Part-Of-Speech Tagger

A special resource for tagging tokens of text with corresponding parts of speech. The Hepple Tagger uses a lexicon and a rule-set, obtained as the result of machine learning on a large corpus taken from the Wall Street Journal. The resource extends the "Token" Annotation with the Feature "pos", which has a value representing a part-of-speech of the current token.

The tagger [Hep00] is a modified version of the Brill tagger, which produces a part-of-speech tag as an annotation on each word or symbol. The tagger uses a default lexicon and rule-set (the result of training on a large corpus taken from the Wall Street Journal). Both of these can be modified manually if necessary. Two additional lexicons exist – one for texts in all uppercase (lexicon cap), and one for texts in all lowercase (lexicon lower). To use these, the default lexicon should be replaced with the appropriate lexicon at load time. The default rule-set should still be used in this case. The ANNIE Part-of-Speech tagger requires the following parameters:

encoding encoding to be used for reading rules and lexicons (init-time).

lexiconURL The URL for the lexicon file (init-time).

rulesURL The URL for the rule-set file (init-time).

document The document to be processed (run-time).

inputASName The name of the annotation set used for input (run-time).

outputASName The name of the annotation set used for output (run-time). This is an optional parameter. If user does not provide any value, new annotations are created under the default annotation set.

baseTokenAnnotationType The name of the annotation type that refers to Tokens in a document (run-time, default = Token)

baseSentenceAnnotationType The name of the annotation type that refers to Sentences in a document (run-time, default = Sentences)

outputAnnotationType POS tags are added as category features on the annotations of type outputAnnotationType (run-time, default = Token)

C.6 Verb Group Chunker

The rule-based verb chunker is based on a number of grammars of English ([Cob99], [Aza89]). 68 rules were developed for the identification of non recursive verb groups. The rules cover finite ('is investigating'), non-finite ('to investigate'), participles ('investigated'), and special verb constructs ('is going to investigate'). All the forms may include adverbials and negatives. The rules have been implemented in JAPE. The finite state analyser produces an annotation of type 'VG' with features and values that encode syntactic information ('type', 'tense', 'voice', 'neg', etc.). The rules use the output of the POS tagger as well as information about the identity of the tokens (e.g. the token 'might' is used to identify modals).

C.7 Multi-lingual Noun Phrase Extractor

MuNPEx is a base NP chunker, i.e., it does not deal with any kind of conjunctions, appositions, or PP-attachments. It was developed by Witte [Wit]. It is implemented in JAPE and can make use of previously detected named entities (NEs) to improve chunking performance.

For each detected NP, an annotation “NP” is added to the document, which includes several features:

DET the determiner of the NP

MOD a list of modifiers of the NP

HEAD the head noun of the NP

Optionally, it can generate additional features indicating the textual positions of the slots described above:

HEAD_START *(optional)* the position in the document where the NP's HEAD starts

HEAD_END *(optional)* the position in the document where the NP's HEAD ends

and similarly for the other slots.

C.8 Fuzzy Coreferencer

The Fuzzy Coreferencer groups the NPs extracted by NPE into coreference chains, ordered sets of NPs that refer to the same entity. It considers definite and indefinite NPs, dates, amounts, and third person pronouns. It is based on a few shallow heuristics which operate on the ordered set of NPs produced by NPE. The different heuristics are distinguished by their likelihood to produce a valid result: string equality is more likely to indicate correct coreference than matching only by head noun. Using fuzzy values allows an explicit representation of the certainty of each stipulated coreference: an NP is assigned to a coreference chain with a certain likelihood. To determine the final coreference chains, the system can now be biased: setting a threshold of 1 for chain membership essentially removes the fuzzy component from the system and results in very short, accurate coreference chains yielding higher precision. Setting a more lenient threshold allows more NPs into the chain, risking false positives but also results in higher recall values. The Fuzzy Coreferencer takes as input the Noun Phrases (or more precisely the *Head* of the NPs) extracted by NPE and attempts to group them into different chains. For every NP in the text, the Fuzzy Coreferencer iterates through the rest of the NPs and checks whether or not NP_1 and NP_n coreferer, i.e., represent the same entity, in similar or different terms. For example, given the following NPs:

- NP_1: Jean Chretien
- NP_2: the Canadian province
- NP_3: meeting
- NP_4: Canada's Prime Minister
- NP_5: he
- NP_6: Quebec

Assuming NP_2 and NP_6 were in an apposition in a text i.e. "the Canadian province, Quebec, ... " they would corefer indicated by the apposition heuristic and belong to the same chain. If they were not in an apposition they might also corefer, depending on other heuristics considering for example the WordNet Semantic Distance, and of course the adjusted merge degree.

Similarly, NP_1 and NP_4 corefer. Also, it is possible that NP_5 gets added to the same chain. The Fuzzy Coreferencer, after clustering the NPs, also attempts to merge the clusters.

As NP_3 does not corefer with any of the listed NPs, it is placed in a chain with itself as the only element.

Details on the fuzzy algorithms are available in [WB03, BWK+03].

Appendix D

Wall Street Journal article from 01.23.87

This shows a complete article from the WSJ dealing with the Iran-Contra affair.

> Fired National Security Council aide Lt. Col. Oliver North regularly discussed his campaign to provide aid to Nicaraguan rebels and his efforts to free U.S. hostages in Lebanon with Attorney General Edwin Meese, Reagan administration sources said.
>
> Justice Department officials have said that Mr. Meese and Col. North regularly met or talked by telephone. "They used to talk on the phone all the time," one official recalled.
>
> The contacts began shortly after President Reagan directed Col. North in late 1984 to help establish a private network to help the rebels, sources said.
>
> Another official said discussions between the two men about the Nicaraguan aid network "are consistent with (Mr. Meese's) dual responsibilities" as attorney general and a member of the NSC. Officials said that even Mr. Meese's closest aides didn't participate in some of the discussions, however, and it isn't known whether Col. North ever told Mr. Meese that profits from secret Iranian arms sales were being diverted to the Contra rebelS.
>
> When he first reported finding evidence of the diversion last Nov. 25, Mr. Meese said Col. North and former National Security Adviser John Poindexter were the only officials who had known of it. A spokesman said Mr. Meese was in San Diego yesterday and wasn't available for comment.
>
> But Reagan administration sources said some U.S. officials tried to warn Adm. Poindexter and other top NSC officials last spring about possible illegalities in the Reagan administration's efforts to help the Nicaraguan rebelS. Adm. Poindexter dismissed the warnings, saying administration lawyers had reviewed Col. North's activities in support of the rebels and had concluded that the Contra aid program was legal, the sources said.
>
> Specifically, intelligence sources said, the CIA station chief in El Salvador became concerned that U.S. military advisers there were assisting a Contra airlift, based at the Ilopango military airfield outside San Salvador and run by retired Air Force Maj. Gen. Richard Secord.
>
> Separately, Reagan administration sources said, some White House officials became alarmed that Col. North apparently had been collecting and disbursing large amounts of cash for the Contras from his office on the third floor of the Old Executive Office Building next to the White House. The sources said Col. North had none of the required government vouchers for the money.

Nevertheless, Reagan administration sources said Adm. Poindexter resisted repeated efforts to force Col. North to share responsibility for his secret Iran and Contra operations with other NSC staff memberS. During the spring of 1986, the sources said, NSC specialists in Central American affairs and in intelligence matters separately became concerned about some of Col. North's activities and asked to be included in overseeing them.

Adm. Poindexter rejected the requests, sources said, and turned down a suggestion from another aide that the NSC staff's Middle East expert, Dennis Ross, be consulted about the Reagan administration's secret Iran policy.

"It was clear that Ollie (Col. North) had someone's hand on his shoulder," said one Reagan administration official. "He was never perceived as an unauthorized loner."

Administration officials said they believe Col. North also kept Vice President Bush and his national security adviser, Donald Gregg, informed about some aspects of the Contra aid network. Mr. Bush and his aides have tried hard to distance themselves from the scandal, but the team headed by independent counsel Lawrence Walsh has indicated it intends to delve into contacts that Mr. Bush's office had with Contra supporters, according to law enforcement officialS.

Among other things, the officials said, Mr. Walsh is expected to focus on why Army Col. Sam Watson, Mr. Bush's deputy national security adviser, received two telephone calls from a former CIA operative in Central America early last October, alerting him that a cargo plane shipping supplies to the Contras was missing.

Justice Department officials didn't interview Mr. Bush or any of his aides during the initial phase of the investigation into the Iran-Contra scandal, law enforcement officials said. Senior Justice Department officials have said they decided that such interviews weren't necessary during the preliminary "fact-finding effort."

Interviews with some of Mr. Bush's aides were conducted after the Federal Bureau of Investigation began a full-scale criminal investigation into the matter at the beginning of December, according to law enforcement officialS. White House officials said Mr. Bush was interviewed by FBI agents for 25 minutes on Dec. 12. Other officials said they believed a wide-ranging interview with Mr. Bush hasn't been scheduled.

Officials said Col. North worked closely with Assistant Secretary of State Elliott Abrams and top CIA officials on the Contra aid network, and frequently traveled around the U.S. and abroad to oversee the ostensibly private network. Col. North often talked twice a day with the CIA's director of operations, Clair George, intelligence sources said.

On several occasions, the sources said, CIA agents who received requests or instructions from Col. North checked with their superiors and were told to do as Col. North directed because he was acting with authority from CIA Director William Casey and Adm. Poindexter.

Mr. Abrams was instrumental in seeking Contra aid from foreign governments, the sources said. In addition to soliciting a $10 million contribution from the sultan of Brunei, Mr. Abrams proposed cabling U.S.

embassies in several Persian Gulf states and asking them to solicit similar contributions, Reagan administration sources said.

That proposal was blocked by the State Department's top Middle East expert, Assistant Secretary Richard Murphy, the sources said.

Reagan administration sources said that while it isn't clear whether Col. North had approval for all his activities, particularly the diversion of Iranian arms-sale profits to the Contras, President Reagan and then-National Security Adviser Robert McFarlane directed him in the fall of 1984 to oversee the Nicaraguan rebels' private aid network.

The private Contra aid operation, approved by Mr. Reagan after meetings with his top advisers, was designed to keep the Contras going until the Reagan administration could overturn a ban on U.S. aid to the rebels, administration officials said. Congress imposed the ban in 1984.

As originally conceived and approved, Reagan administration sources said, the plan called for the president and other top officials to attend fund-raising events for the rebels, to invite Contra leaders to well-publicized White House meetings, and to encourage private donors to support the anti-Sandinista rebelS. "It was all fairly overt," said one Reagan administration official.

But according to officials, Col. North began ranging farther and farther afield as it became clear that private donors couldn't keep pace with the Contras' needs – or with the Sandinistas' Soviet-supported military buildup.

A Danish shipping agent yesterday confirmed that a freighter leased by his firm carried arms for the insurgents from Portugal to Honduras as early as mid-1985.

Mr. Parlow indicated that other Danish ships helped carry arms to Honduras, and he said the freight carried in 1985 was paid for through Defex-Portugal, a Lisbon firm that was instrumental in later arms shipments to the ContraS.

Though denying any role in the Reagan administration's Iran arms sales, Mr. Parlow also confirmed that the freighter Erria had been off the coast of Cyprus late last May, when intelligence sources have said it was part of an unsuccessful effort by Col. North to ransom U.S. hostages in Lebanon.

Mr. Parlow also said he is now the Erria's owner, having purchased the vessel last April for an estimated 2.5 million Danish kroner – about $350,000 at current exchange rates – and he confirmed that he has been a client of Willard Zucker, an attorney at CSF and a central figure in the arms network.

White House spokesman Larry Speakes said yesterday that no tape or transcript will be made of Mr. Reagan's meeting with the Tower Commission on Monday, nor will notes of the session be released. Mr. Speakes said the commissioners felt that transcribing the encounter would demean the presidency.

Appendix E

System Example Output

We give an example of the system processing a couple of real newspaper articles. After presenting the relevant excerpts fo the articles, we show excerpts of the system output.

E.1 Selection of excerpts from WSJ articles dealing with Iran-Contra Affair

Here we present excerpts from newspaper articles related to the Iran-Contra affair published in the Wall Street Journal between 01.07.87 and 05.04.87.

WSJ 01/07/87 Some congressional sources say Mr. McFarlane's account rests on a personal conversation with Mr. Reagan that Mr. Meese and Mr. Regan may not know about, and that the president may have forgotten.

Both Mr. Regan and the vice president say they knew nothing of the diversion.

Lt. Col. North has reportedly said the Israelis came up with the funds-diversion idea. But Israel denies that it knew anything about the diversion.

Contra leader Adolfo Calero insists the rebels never saw any of the millions of dollars supposedly sent their way by Lt. Col. North.

WSJ 01/09/87 Mr. Speakes said a small group of administration officials that reviewed the report had urged the deletion of details concerning two contacts with Israel.

Mr. Speakes suggested that the president's letter and the reference to the vice president's meeting were suppressed for "diplomatic" rather than political reasonS.

WSJ 01/30/87 A Senate Intelligence Committee report on the Iran-Contra affair indicates that the diversion of profits to Nicaraguan guerrillas was an early feature of the secret U.S. arms sales to Iran.

The report provides no evidence that President Reagan knew of or approved the diversion of fundS.

According to the Senate report, the meetings came amid increasing concern in the administration about financial support for the ContraS.

The next day, according to documents received by the Senate committee, the president and his top foreign policy advisers discussed soliciting "non-lethal" aid for the Contras from other countrieS. According to the documents, Secretary of State George Shultz was to provide a list of countries that could be approached.

The report suggests President Reagan was deeply involved in the unfolding policy toward Iran.

The report says that Mr. Reagan was briefed about the arms-hostages link.

Mr. Regan testified that the president was displeased to learn of the Israeli transfers after the fact and instructed his staff "to let the Israelis know of our displeasure."

According to the Senate report, a CIA lie detector test given to Mr. Ghorbanifar last January indicated he could be an Iranian agent who was trying to deceive the U.S.

Mr. Meese told the committee that the day after he began his initial efforts to unravel the trail of money, he held a Saturday morning meeting with Mr. Shultz at the secretary's suburban Washington home, according to the report.

Meanwhile, Mr. Meese, according to the committee's narrative, gave an opinion last January that withholding congressional notification of the arms sales was legal.

WSJ 02/05/87 In fact, sources said, officials concluded then that weapons the network was providing were overpriced and of poor quality.

Administration officials said last autumn that they were generally aware of the private supply operation, but didn't know many details about it.

Administration sources said Col. North's boss, then National Security Adviser John Poindexter, became concerned about Col. North's Contra aid network in August and tried to ease Col. North out of his role as overseer of the administration's Central American policieS.

Also, the sources said, the administration's new interagency group to oversee the Contra aid program didn't include Col. North.

Administration sources said Col. North pushed hard to be included in the new group and eventually succeeded.

Officials said the private arms network, overseen by Col. North and retired Air Force Maj. Gen. Richard Secord, continued operating even after the administration created its own Contra-supply mechanism, partly because congressional delays in approving a new budget forced the CIA to postpone setting up its own network.

WSJ 02/18/87 Mr. Gates said yesterday that he was "uneasy" when Col. North referred to Swiss accounts and the Contra funding at a luncheon with Mr. Casey two days later, on Oct. 9.

Mr. Gates indicated that senior CIA officials were worried about the private individuals assisting Col. North in the Iran and Contra operationS.

In December he said that Col. North alluded to a Swiss account and money for the Contras in the context of a discussion related to the financial disarray and problems facing an Iranian middleman involved in the arms sales to Tehran. Yesterday, the acting director said the reference had come only in a discussion related to the downing of a supply plane in Nicaragua, and Mr. Gates hadn't associated the remark with "the Iranian matter at all."

WSJ 02/25/87 A 1986 memo detailing proposed diversion of profits from secret arms sales to Iran to help Nicaraguan insurgents was written by Lt. Col. Oliver North to his boss, then-National Security Adviser John Poindexter, law enforcement and Reagan administration officials said.

The suit also contends that all of the investigators appointed by Mr. Walsh are serving without authority, and it seeks an injunction blocking all further investigation by the independent counsel.

Through a spokeswoman, Mr. Walsh said, "The independent-counsel statute has received extensive and careful study, and we are satisfied of its constitutionality."

Law-enforcement officials said allegations that certain White House documents were altered or destroyed have raised new questions about why Attorney General Edwin Meese waited several days after he first became suspicious that arms-sale funds were being diverted to the Contras before launching a full-scale criminal probe of the matter.

Administration officials said Adm. Poindexter kept some sensitive documents, including the only copy of a January 1986 presidential directive authorizing U.S. arms sales to Iran, in a safe in his office. The officials said they believe Paul Thompson, a lawyer on the NSC staff, also had access to the safe, which they said Adm. Poindexter cleaned out after he resigned last Nov. 25.

Mr. Meese has said that when he began looking into the arms-sales operation four days earlier, he didn't think it was necessary to take steps to safeguard documents or try to keep witnesses from comparing versions of their stories because "there was no indication or suggestion of any criminality."

WSJ 02/27/87 Reagan administration officials secretly channeled extensive military aid to Nicaraguan insurgents during a two-year ban on U.S. arms assistance for the rebels, a presidential commission concluded.

The panel's report constitutes the strongest official confirmation that ranking officials in the Reagan administration sought to circumvent the ban by mounting a private war against Nicaragua's government, and repeatedly misled Congress about it.

The findings suggest that laws may have been breached, though the commission carefully avoided reaching conclusions on legal issueS.

Marine Lt. Col. Oliver North, the fired NSC aide, helped to oversee a supply network that boasted millions of dollars in private assets, including planes, warehouses and a landing strip in Costa Rica, according to the report.

But last May, when the ban on U.S. arms assistance remained in effect, Col. North privately indicated he was acting with the president's knowledge, according to a memo cited in the report. "The president obviously knows why he has been meeting with several select people to thank them for their 'support for Democracy' in CentAm.," Col. North wrote to his superior, Vice Adm. John Poindexter, then the national security adviser.

But Adm. Poindexter, who resigned when the scandal broke, was fearful in the same period that details about Col. North's role might become known to Congress or the public, NSC memos show.

Assisting Col. North, the commission says, was a network of private individuals who would also play a role in shipping U.S. arms to Iran. Gen. Secord was provided a special encrypting device from the National Security Agency - the government's electronic eavesdropping agency - to allow him to better communicate secretly with Col. North, the report sayS.

Last July, after Congress was moving toward approval of renewed military aid to the Contras, Col. North estimated the total assets of Project Democracy at more than $4.5 million, according to the report. He proposed that the CIA purchase the equipment, some of which, he indicated, had been bought from the U.S. Air Force originally under special "proprietary arrangements."

WSJ 03/02/87 A secret Contra supply operation received an infusion of funds after U.S. weapon sales to Iran last May, according to White House documents that shed new light on the multilayered financial network used to sustain the Nicaraguan guerrillaS.

At least two bank accounts identified in Col. North's papers are believed to be controlled by Contra leader Adolfo Calero, and a third, in Costa Rica, was the source of payments to Contra official Alfonso Robelo, according to intelligence sourceS.

Though Col. North prepared a memo last April outlining plans to divert an estimated $12 million to support the Contras, sources familiar with the Tower Commission investigation said they found no evidence that the document was shown to President Reagan.

More generally, however, Col. North's tone assumes that Mr. Reagan had some knowledge regarding the Contra supply operation, and investigators say the Marine officer was prolific in the detail he provided his immediate superior, Vice Adm. Poindexter. In one message, he suggests that Col. North join him in retirement and they both could become involved in clandestine activitieS.

Col. North reported an increase of $16.5 million in the funding available to the insurgents between Feb. 22 and April 9, 1985.

WSJ 03/16/87 The administration's policy says it supports the Contras to pressure the Sandinista government into negotiations with its domestic political opponents and neighboring countrieS.

The administration's opponents say the Sandinistas use the civil war against the Contras as an excuse to avoid negotiations on relaxing domestic political restrictionS.

The administration's critics also say the Contras are a corrupt fighting force whose leaders have as little interest in bringing democracy to Nicaragua as the Sandinistas appear to.

WSJ 03/19/87 Committee members said the decision to focus the first hearings on the U.S. role in Nicaragua was dictated in part by the chronology of eventS. But it also has political importance because it will influence how the entire Iran-Contra affair is cast in the months ahead.

"The diversion of funds from the Iran arms sales is only a part of the puzzle, and maybe a very small part," a congressional source said. "We first want to focus on how the private network which supplied the Contras got set up in 1984, and whether (President) Reagan authorized it."

Intelligence and congressional sources said William Casey, former Central Intelligence Agency director, appears to have been one of the architects of the private aid program. They said Mr. Casey directed some of his subordinates to help Lt. Col. Oliver North, fired NSC aide, supply arms to Iran and aid to the ContraS.

Separately, intelligence sources said a report by the CIA's inspector general concluded that officials at CIA headquarters in Langley, Va., and at CIA stations in Europe and Central America helped funnel aid to the Contras despite a CIA directive to avoid violating the congressional ban at that time on aid to the insurgentS.

WSJ 03/26/87 The sources said Mr. McFarlane also reported to the president regularly on secret arms shipments to the rebels in 1985 and on efforts by private groups in the U.S. to raise money for the ContraS. "McFarlane told the president everything he knew," said one former senior official.

But the officials said it isn't clear whether Mr. Reagan, Mr. McFarlane, and other top officials knew that fired National Security Council aide Lt. Col. Oliver North was controlling the secret arms network, directing arms shipments to the Contras, and providing crucial military intelligence to the rebelS. "This was all presented as things the Contras were doing for themselves, not as things Ollie (Col. North) was doing for them," said one former senior official.

And there still is no evidence, administration and congressional sources said, that the president knew profits from U.S. arms sales to Iran were diverted to the Contras in 1986. Mr. Reagan has repeatedly said he knew nothing of any diversion before Justice Department officials found evidence of it in Col. North's files last November.

President Reagan said at his news conference last week that he knew "that there were many people privately giving money" to aid the Contras during the period Congress had cut off U.S. funds to them.

The president said he didn't believe such donations were illegal because they were private. But according to current and former administration officials and some Contra fund-raisers, Mr. Reagan, Mr. McFarlane and other officials believed the secret aid program was legal in part because Col. North misled his superiors about the extent of his involvement with the rebelS.

Mr. McFarlane, according to current and former officials, told Col. North several times in 1985 that he couldn't become a broker for aid to the Contras or get into the business of actually buying and selling weaponS. Nevertheless, according to a report by the presidential commission that investigated the Iran-Contra affair, Col. North quickly became the chief executive of a secret Contra aid network whose assets he later estimated at $4.5 million.

Mr. McFarlane, three former officials said, then drafted a letter to former Rep. Michael Barnes (D., Md.) insisting that the NSC staff hadn't been supporting the Contras "directly or indirectly" and gave the letter and the documents collected from Col. North to then-White House counsel Fred Fielding. According to Mr. Fielding, Mr. McFarlane never asked him to review the legality of Col. North's activitieS.

Current and former officials and other sources familiar with the Contra aid network say Mr. Casey and other U.S. officials began soliciting contributions to the Contras and other anti-communist causes from Saudi Arabia, Israel and other nations as early as 1982.

In late 1983, intelligence sources said, the CIA bought a large supply of Soviet-bloc weapons, including automatic rifles and shotguns, which Israel had captured from the Palestine Liberation Organization in Lebanon the previous year. The weapons were transferred to CIA warehouses, then distributed to a variety of covert operations, including the effort to unseat Nicaragua's Sandinista regime, the sources said.

When the Saudi Arabians offered to help finance the Contras in mid-1984, the president and his top advisers still hadn't debated the merits of seeking outside help for the insurgents, according to a former senior official deeply involved in the Contra aid issue.

WSJ 04/07/87 Stephen Trott, the associate attorney general and one of Mr. Meese's closest advisers, maintains that "Ed Meese deserves a medal" for initially uncovering suspected diversion of funds to the ContraS.

Congressional investigators say that Mr. Webster and one of his top aides, Oliver "Buck" Revell, knew more about the secret arms shipments to Iran - and learned about them months earlier - than previously disclosed.

Mr. Meese maintains that he never provided any legal advice or violated any congressional bans related to Col. North's efforts to help ship weapons and other supplies to the ContraS.

WSJ 04/22/87 Moreover, the investigators said that Adm. Poindexter should be able to say whether President Reagan approved the creation of the private network to aid the Contras and whether Mr. Reagan authorized or knew about the National Security Council's role in directing it.

Meanwhile, law-enforcement officials said Lawrence Walsh, the independent counsel investigating the Iran-Contra affair, has begun looking into reports that a secret Army anti-terrorist unit may have set up a numbered Swiss bank account in 1983. That account, according to a report first aired by CBS News, was used later by Gen. Secord and Lt. Col. Oliver North to funnel aid to the Contras in violation of congressional restrictions on such assistance.

Pentagon spokesmen declined to comment on the reports, except to say that neither Defense Secretary Caspar Weinberger nor Secretary of the Army John Marsh approved any improper fund transferS. Officials said that Mr. Walsh, among other things, wants to determine if Col. North, the fired White House aide, ever controlled the account.

WSJ 04/29/87 Retired Air Force Maj. Gen. Richard Secord, a central figure in the Iran-Contra affair, has agreed to testify before congressional investigating committees voluntarily and without immunity from prosecution, according to people familiar with the inquiry.

Congressional investigators said Gen. Secord is a crucial witness because he helped arrange the U.S. arms deals with Iran and set up a private air resupply network for the Nicaraguan rebels known as the ContraS.

Mr. Walsh previously hadn't asserted that any government funds used in the Iran-Contra affair were missing.

Separately, President Reagan said John Poindexter, the former White House national security adviser, never told him about diversion of proceeds from the Iran arms sales to the ContraS. "Maybe he thought he was being, in some way, protective of me," Mr. Reagan told a group of reporters, adding that "I did not know that there were any excess funds" until "after the whole thing blew up."

WSJ 04/30/87 But his courtroom allegation concerning Col. North, combined with comments by criminal investigators, indicate that independent counsel Lawrence Walsh is attempting to build a broader conspiracy case - partly by relying on cooperation from Mr. Channell and three employees of the fund-raiser who weren't identified.

Mr. Channell named Col. North in open court as part of his plea agreement with the special prosecutor, individuals familiar with Mr. Walsh's investigation said.

Law enforcement officials said Mr. Walsh also has been investigating whether Mr. Channell, or leaders of other pro-Contra groups, may have violated federal laws barring U.S. citizens from participating in foreign military conflictS.

WSJ 05/01/87 The president insists that he didn't trade arms for Americans held hostage in Lebanon and that he didn't know that money from his administration's secret arms sales to Iran was diverted to help Nicaraguan rebelS.

"Reagan deeply believed that the Contras were an important element in the security of America," says Sen. Warren Rudman, the New Hampshire Republican and vice chairman of the Senate investigating committee. Adds a senior congressional investigator: "He made the decisions, and he wasn't manipulated. He made the fundamental decision that the Contras shouldn't perish."

Congressional investigators say Mr. Reagan's personal diary, his appointment calendar and his public statements all show he was actively involved in raising funds for the Contras and in encouraging his aides to do likewise. But, they say, it isn't clear that the president's advisers told him about everything they were doing. And investigators say they don't yet have any evidence that Mr. Reagan knew profits from the sale of arms to Iran were diverted to help the Contras, or that he approved any law-breaking by his aides or otherS.

Rather, say congressional investigators, the panels want to know whether the president entrusted two of his most important foreign-policy priorities - Iran and the Contras - to a handful of privateers in a deliberate effort to flout congressional and legal restraints on covert action and aid to the Nicaraguan insurgentS.

The Tower board concluded that the president's detached management style allowed overzealous aides to draw the U.S. into ill-advised and embarrassing secret dealings with Iran and the ContraS. The new hearings, however, will show that the president was actively involved in the policies, and not a "befuddled" bystander, predicts Sen. Daniel Inouye, the Hawaii Democrat who heads the Senate panel.

Senior White House officials concede that it may be difficult to prove the president didn't know about and didn't authorize a diversion.

According to someone familiar with the congressional investigation, retired Air Force Col. Robert Dutton, who ran the aerial resupply operation for Gen. Secord last year, provided photographs for Col. North to use in briefing the president on the operation.

Administration officials say that Adm. Poindexter kept detailed notes of his meetings with President Reagan and, like Mr. McFarlane, briefed the president regularly on aid to the ContraS.

Investigators say it isn't clear whether Col. North knew of the Saudi contributions or whether his memo was designed to conceal the existing Saudi aid.

In an effort to guarantee the Contras' survival, Reagan officials and people familiar with the congressional investigation say, Col. North turned to Gen. Secord to help organize a more effective private aid network.

"Getting aid to the Contras was a preoccupation of Reagan, Regan and Casey," says one senior intelligence source. "Reagan certainly knew that Ollie North was in charge of efforts to get help for the Contras."

While Gen. Secord and associates assembled a complex network of companies, bank accounts, and individuals to funnel weapons and supplies to the rebels, Mr. Casey personally recruited the CIA station chief in Costa Rica to assist them, intelligence sources say.

Congressional investigators say that funds diverted from the Iranian arms sales and other moneys were routed through Swiss bank accounts and a network of companies in the U.S., Europe and Central America.

Gen. Secord has told friends that he didn't profit from his involvement in the affair.

WSJ 05/04/87 Meanwhile, two persons familiar with the congressional investigation said that retired Air Force Maj. Gen. Richard Secord, who is a central figure in the Iran-Contra affair and is scheduled to be the lead-off witness, is prepared to testify that neither he nor Col. North were signatories on Swiss bank accounts set up in the name of Lake Resources Inc., a Panamanian-registered company.

In another development, a spokesman for Joseph Coors said that the prominent conservative and staunch Reagan backer contributed $60,000 to help the Contras after Col. North solicited his aid in 1985. In response to questions, Robert Walker, a vice president of Adolph

Coors Co.'s Washington office, said "North assured us at the time that all the legal research had been done on this and it was absolutely legal."

Mr. Walker, who said that he and Mr. Coors had recently been interviewed by the FBI, said the donation was made for medical supplieS.

But Mr. Webster said that Col. North's actions didn't impede any investigationS.

While some of the messages apparently never got to the director or his senior staff because of transmission problems, Mr. Revell said, there were enough other hints of questionable activities by Col. North to warrant further investigation by lower-level FBI officialS.

On another matter, Sen. Daniel Inouye (D., Hawaii), chairman of the Senate committee, Sen. Rudman and another congressional investigator said there isn't any doubt that President Reagan knew money was being raised to help the Contras during the period when Congress had cut off U.S. aid. But they said it isn't clear that Mr. Reagan knew whether the administration was participating in that effort.

But the chairman also said Mr. Reagan's diaries show the president "wasn't particularly knowledgeable about some of the details."

While traveling from Washington to New York yesterday, Mr. Reagan told reporters he was "aware that there were people" raising private funds to help the Contras, "but there was nothing in the nature of a solicitation by the administration, to my knowledge, of anyone to do that." He also said he had "no detailed information" that the money was used for weaponS.

Separately, it was reported that committee investigators have added some new details to earlier disclosures that the covert arms network supplying the Contras had obtained weapons in Poland and China. In February, The Wall Street Journal disclosed that a ship used to carry arms to the Contras had also transported Soviet-bloc weapons from Poland to a Pentagon warehouse in North Carolina for later distribution to anti-Communist guerrillaS. The New York Times reported Saturday that committee staffers have evidence that the Polish weapons were bought for the ContraS.

In addition, the Times said, the committee has evidence that weapons sold to the Contras by China included Soviet-made anti-aircraft missileS.

E.2 Fuzzy Believer output for Iran-Contra Affair articles

The following pages show a subset of the original output of the Fuzzy Believer system for the articles above. Note, that statements can be part of more than one domain and that one statement can have more than one predicate-argument structure. The reporter is considered the same for all articles to not confuse the reader with too much information. The example shows the *majority strategy*, for which the reporter information is not relevant.

Figure E.1: System example output (a)

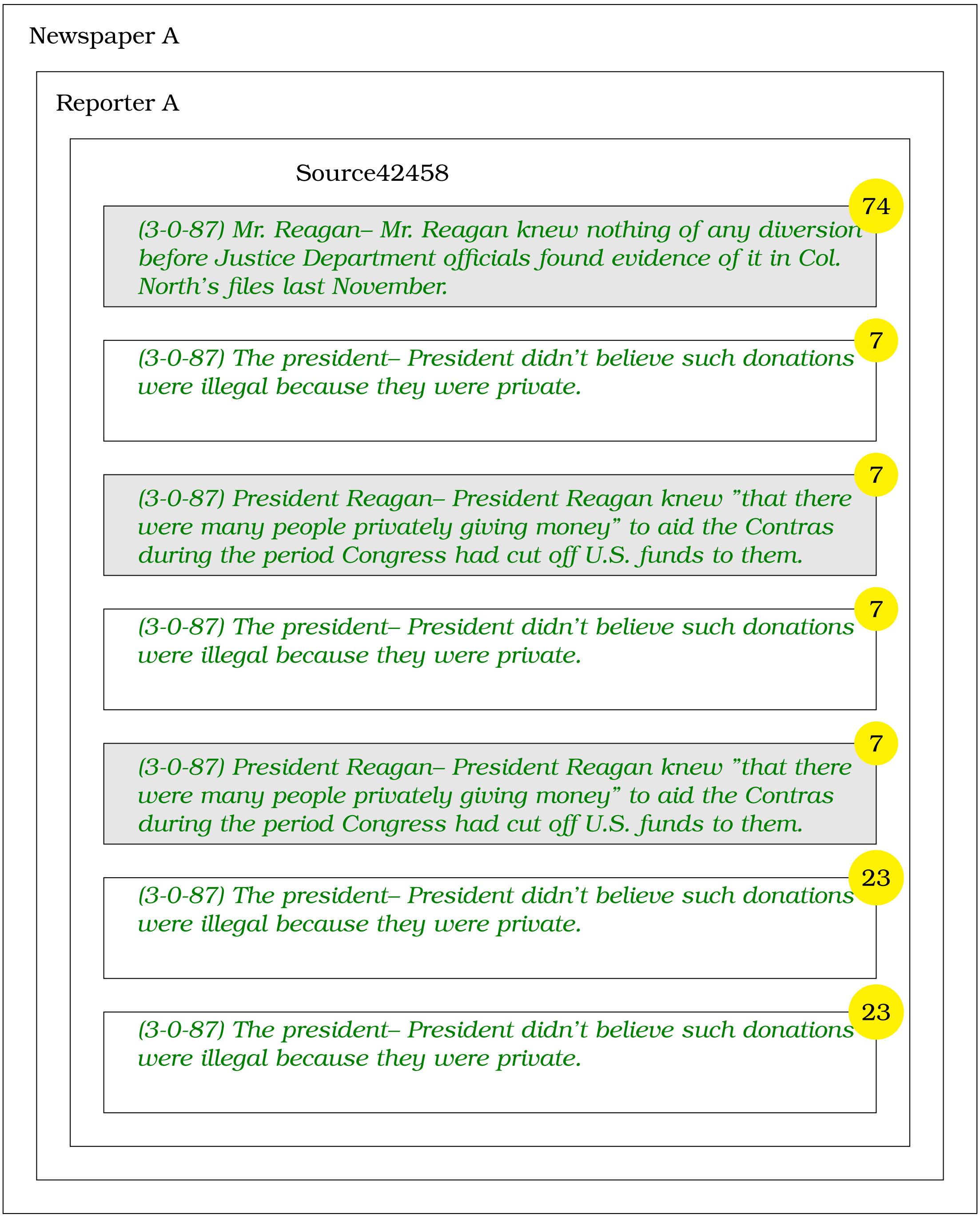


Figure E.2: System example output (b)

Newspaper A

Reporter A

Source43499

Source43216

(4-5-87) President Reagan– John Poindexter, the former White House national security adviser, never told John Poindexter about diversion of proceeds from the Iran arms sales to the Contras. 11

Source42458

(3-0-87) The president– President didn't believe such donations were illegal because they were private. 10

(3-0-87) President Reagan– President Reagan knew "that there were many people privately giving money" to aid the Contras during the period Congress had cut off U.S. funds to them. 10

(3-0-87) The president– President didn't believe such donations were illegal because they were private. 10

(3-0-87) President Reagan– President Reagan knew "that there were many people privately giving money" to aid the Contras during the period Congress had cut off U.S. funds to them. 10

(3-0-87) Mr. Reagan– Mr. Reagan knew nothing of any diversion before Justice Department officials found evidence of it in Col. North's files last November. 10

Figure E.3: System example output (c)

Newspaper A

Reporter A

Source42252

(2-5-87) a memo– But last May, when the ban on U.S. arms assistance remained in effect, Col. North privately indicated Col. North was acting with the president's knowledge, according to.

58

(2-5-87) a memo– But last May, when the ban on U.S. arms assistance remained in effect, Col. North privately indicated Col. North was acting with the president's knowledge, according to.

1

Source41967

(2-1-87) The next day,– Documents received by the Senate committee, the president and his top foreign policy advisers discussed soliciting "non-lethal" aid for the Contras from other countries.

5

Source42922

(5-1-87) Gen. Secord– Friends that Gen. Secord didn't profit from his involvement in the affair.

19

Source43209

(5-4-87) The New York Times– Saturday that committee staffers have evidence that the Polish weapons were bought for the Contras.

10

(5-4-87) The New York Times– Saturday that committee staffers have evidence that the Polish weapons were bought for the Contras.

7

Figure E.4: System example output (d)

Newspaper A

Reporter A

Source43480

(5-1-87) Congressional investigators– Mr. Reagan's personal diary, his appointment calendar and his public statements all show Mr. Reagan's was actively involved in raising funds for the Contras and in encouraging his aides to do likewise. 17

(5-1-87) investigators– They don't yet have any evidence that Mr. Reagan knew profits from the sale of arms to Iran were diverted to help the Contras, or that Mr. Reagan approved any lawbreaking by his aides or others. 27

(5-1-87) Investigators– It isn't clear whether Col. North knew of the Saudi contributions or whether his memo was designed to conceal the existing Saudi aid. 12

(5-1-87) a senior congressional investigator– Vice chairman made the fundamental decision that the Contras shouldn't perish. 3

(5-1-87) a senior congressional investigator– Vice chairman made the decisions, and vice chairman wasn't manipulated. . 3

(5-1-87) Investigators– It isn't clear whether Col. North knew of the Saudi contributions or whether his memo was designed to conceal the existing Saudi aid. 1

(5-1-87) investigators– They don't yet have any evidence that Mr. Reagan knew profits from the sale of arms to Iran were diverted to help the Contras, or that Mr. Reagan approved any lawbreaking by his aides or others. 1

Figure E.5: System example output (e)

Newspaper A

Reporter A

Source42005

7

(2-1-87) Meese– The committee that tagent day after he began his initial efforts to unravel the trail of money, agent held a Saturday morning meeting with Mr. Shultz at the secretary's suburban Washington home, according to the report.

Source43722

0

(1-6-87) Contra leader Adolfo Calero– The rebels never saw any of the millions of dollars supposedly sent their way by Lt. Col. North.

Source43248

76

(3-4-87) The administration's critics– The Contras are a corrupt fighting force whose leaders have as little interest in bringing democracy to Nicaragua as the Sandinistas appear to.

7

(3-4-87) The administration's critics– The Contras are a corrupt fighting force whose leaders have as little interest in bringing democracy to Nicaragua as the Sandinistas appear to.

7

(3-4-87) The administration's critics– The Contras are a corrupt fighting force whose leaders have as little interest in bringing democracy to Nicaragua as the Sandinistas appear to.

Source43533

5

(2-5-87) The findings– Laws may have been breached, though the commission carefully avoided reaching conclusions on legal issues.

Figure E.6: System example output (f)

Bibliography

[Arg05] 2005.

[Aus62] J. A. Austin. *How to Do Things with Words.* Harvard University Press, 1962.

[Aza89] S. Azar. *Understanding and Using English Grammar.* Prentice Hall Regents, 1989.

[BC06] Ted Briscoe and John Carroll. Evaluating the accuracy of an unlexicalized statistical parser on the parc depbank. In *ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, Sydney, Australia, 17-21 July 2006.* The Association for Computer Linguistics, 2006.

[BCW06] E. Briscoe, J. Carroll, and R. Watson. The Second Release of the RASP System. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, 2006.

[BDGW04] Sabine Bergler, Monia Doandes, Christine Gerard, and René Witte. Attributions. In Qu et al. [QSW04], pages 16–19.

[Ber92] Sabine Bergler. *The Evidential Analysis of Reported Speech.* PhD thesis, Brandeis University, Massachusetts, USA, 1992.

[Ber93] Sabine Bergler. Semantic dimensions in the field of reporting verbs. In *Proceedings of the Ninth Annual Conference of the University of Waterloo Centre for the New Oxford English Dictionary and Text Research*, Oxford, England, September 1993.

[Ber05] Sabine Bergler. Conveying attitude with reported speech. In James C. Shanahan, Yan Qu, and Janyce Wiebe, editors, *Computing Attitude and Affect in Text: Theory and Applications.* Springer Verlag, 2005.

[BHDD+06] Roy Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. The Second PASCAL Recognising Textual Entailment Challenge. In *Proc. of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, 2006.

[BM05] Johan Bos and Katja Markert. Recognising textual entailment with logical inference. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 628–635, Vancouver, British Columbia, Canada, October 2005. Association for Computational Linguistics.

[Boy06] Igor Boyko. Formal semantics of verbs for knowledge inference. In ICoS-06 [ICo06].

[BW91] Afzal Ballim and Yorick Wilks. *Artificial Believers: The Ascription of Belief.* Lawrence Erlbaum Associates, Inc., Mahwah, NJ, USA, 1991.

[BWK+03] Sabine Bergler, René Witte, Michelle Khalife, Zhuoyan Li, and Frank Rudzicz. Using Knowledge-poor Coreference Resolution for Text Summarization. In *Workshop on Text Summarization at HLT-NAACL 2003*, Document Understanding Conference (DUC), Edmonton, Canada, May 31–June 1 2003. NIST.

[BYT+04] Steven Bethard, Hong Yu, Ashley Thornton, Vasileios Hatzivassiloglou, and Dan Jurafsky. Automatic extraction of opinion propositions and their holders. In Qu et al. [QSW04], pages 20–27.

[CC04] Yuan Yan Chen and Joseph J. Chen. Neural networks and belief logic. In *HIS '04: Proceedings of the Fourth International Conference on Hybrid Intelligent Systems (HIS'04)*, pages 460–461, Washington, DC, USA, 2004. IEEE Computer Society.

[CCS05] Marcela Capobianco, Carlos Iván Chesñevar, and Guillermo Ricardo Simari. An argument-based framework to model an agent's beliefs in a dynamic environment. In Rahwan et al. [Arg05], pages 95–110.

[CMB+06] Hamish Cunningham, Diana Maynard, Kalina Bontcheva, Valentin Tablan, Cristian Ursu, Marin Dimitrov, Mike Dowman, Niraj Aswani, and Ian Roberts. *Developing Language Processing Components with GATE Version 4 (a User Guide)*. University of Sheffield, February 2006. For GATE version 3.1.

[CMBT02] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proc. of the 40th Anniversary Meeting of the ACL*, 2002.

[CMT00] H. Cunningham, D. Maynard, and V. Tablan. JAPE: a Java Annotation Patterns Engine (Second Edition). Research Memorandum CS–00–10, Department of Computer Science, University of Sheffield, November 2000.

[Cob99] C. Cobuild, editor. *English Grammar*. Harper Collins, 1999.

[Con03] The Unicode Consortium. *The Unicode Standard, Version 4.0*. Addison-Wesley, Reading, MA, USA, 2003.

[DBQ05] Bill Dolan, Chris Brockett, and Chris Quirk. Mirosoft research paraphrase corpus. Online: `http://research.microsoft.com/research/nlp/msr_paraphrase.htm`, March 2005.

[DGM05] Ido Dagan, Oren Glickman, and Bernardo Magnini. The PASCAL Recognising Textual Entailment Challenge. In *Proc. of the PASCAL Challenges Workshop on Recognising Textual Entailment*, 2005.

[Doa03] Monia Doandes. Profiling For Belief Acquisition From Reported Speech. Master's thesis, Concordia University, Montréal, Québec, Canada, 2003.

[DQB04] Bill Dolan, Chris Quirk, and Chris Brockett. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of Conference on Computational Linguistics (COLING-04)*, pages 350–356, Geneva, Switzerland, 2004.

[Fel98] Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[GACOR05] Michael Gamon, Anthony Aue, Simon Corston-Oliver, and Eric K. Ringger. Pulse: Mining customer opinions from free text. In A. Fazel Famili, Joost N. Kok, José Manuel Peña, Arno Siebes, and A. J. Feelders, editors, *Advances in Intelligent Data Analysis VI, 6th International Symposium on Intelligent Data Analysis, IDA 2005, Madrid, Spain, September 8-10, 2005, Proceedings*, volume 3646 of *LNCS*, pages 121–132. Springer, 2005.

[Ger00] Christine Gerard. Modelling Readers Of News Articles Using Nested Beliefs. Master's thesis, Concordia University, Montréal, Québec, Canada, 2000.

[GHS+05] R. Gaizauskas, M. Hepple, H. Saggion, M. A. Greenwood, and K. Humphreys. SUPPLE: A practical parser for natural language engineering applications. In *Proc. of the 9th Intl. Workshop on Parsing Technologies (IWPT2005)*, Vancouver, 2005.

[Gä88] Peter Gärdenfors. *Knowledge in Flux: Modeling the Dynamics of Epistemic States*. MIT Press, 1988.

[Hep00] Mark Hepple. Independence and commitment: Assumptions for rapid training and execution of rule-based pos taggers. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL-2000)*, Hong Kong, October 2000.

[HLO06] Sangyun Hahn, Richard Ladner, and Mari Ostendorf. Agreement/disagreement classification: Exploiting unlabeled data using contrast classifiers. In HLT/NAACL-06 [HLT06], pages 53–56.

[HLT06] New York City, USA, June 2006. Association for Computational Linguistics.

[Hyd05] Dominic Hyde. Sorites paradox. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Fall 2005.

[ICo06] Buxton, UK, April 2006.

[JM00] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2000.

[KH06] Soo-Min Kim and Eduard Hovy. Identifying and analyzing judgment opinions. In HLT/NAACL-06 [HLT06], pages 200–207.

[Kle06] Beata Beigman Klebanov. Measuring Semantic Relatedness Using People and WordNet. In HLT/NAACL-06 [HLT06], pages 13–16.

[KS00] Aaron N. Kaplan and Lenhart K. Schubert. A computational model of belief. *Artificial Intelligence*, 120(1):119–160, 2000.

[LF06] Boon Toh Low and Norman Y. Foo. Consistency reasoning in neural-logic belief networks. Technical Report TR93-467, Basser Department of Compurter Science, University of Sydney,NSW, 2006.

[LG91] Douglas B. Lenat and R. V. Guha. The evolution of cycl, the cyc representation language. *SIGART Bull.*, 2(3):84–87, 1991.

[LH02] Shian-Hua Lin and Jan-Ming Ho. Discovering informative content blocks from web documents. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, Edmonton, Alberta, Canada, July 2002.

[Lid01] E.D. Liddy. *Natural Language Processing*. Marcel Decker, Inc, NY, 2nd edition, 2001.

[Lin94] Dekang Lin. Principar: an efficient, broad-coverage, principle-based parser. In *Proceedings of the 15th conference on Computational linguistics*, pages 482–488, Morristown, NJ, USA, 1994. Association for Computational Linguistics.

[Lin98] Dekang Lin. Dependency Based Evaluation of MINIPAR. In *Proceedings of the Workshop on the Evaluation of Parsing Systems, First International Conference on Language Resources and Evaluation*, 1998.

[Mar01] Victor V. Martynov. *Summary*. European Humanity University, 2001.

[Mat04] Yvette Yannick Mathieu. Linguistic knowledge and automatic semantic representation of emotions and feelings. In *International Conference on Information Technology: Coding and Computing (ITCC'04), Volume 2, April 5-7, 2004, Las Vegas, Nevada, USA*, pages 314–318. IEEE Computer Society, 2004.

[MGdM+06] Bill MacCartney, Trond Grenager, Marie-Catherine de Marneffe, Daniel Cer, and Christopher D. Manning. Learning to recognize features of valid textual entailments. In HLT/NAACL-06 [HLT06], pages 41–48.

[Min68] Marvin Minsky. Matter, mind, and models. pages 425–432, 1968. From Minsky (ed), Semantic Information Processing.

[Min75] Marvin Minsky. A framework for representing knowledge. Ai memo, MIT, 1975. Excerpt:Winston (ed), Psychology of Computer Vision, 1975, haugeland81,luger95.

[MK96] John Maxwell and Ron Kaplan. An efficient parser for lfg. In *First LFG Conference*, Grenoble, France, 1996.

[NCK06] Rowan Nairn, Cleo Condoravdi, and Lauri Karttunen. Computing relative polarity for textual inference. In ICoS-06 [ICo06].

[NØ05] Kjetil Nørvåg and Randi Øyri. News item extraction for text mining in web newspapers. In *Proceedings of International Workshop on Challenges in Web Information Retrieval and Integration (in conjunction with ICDE'2005)*, Tokyo, Japan, April 2005. IEEE Computer Society.

[PC05] Fabio Paglieri and Cristiano Castelfranchi. Revising beliefs through arguments: Bridging the gap between argumentation and belief revision in mas. In Rahwan et al. [Arg05], pages 78–94.

[PHW+03] S. Pradhan, K. Hacioglu, W. Ward, J. Martin, and D. Jurafsky. Semantic role parsing: Adding semantic structure to unstructured text. In *Proceedings of the International Conference on Data Mining (ICDM-03)*, 2003.

[Por13] Noah Porter, editor. *Webster's Revised Unabridged Dictionary*. G & C. Merriam Co., 1913.

[QSW04] Yan Qu, James Shanahan, and Janyce Wiebe, editors. Technical Report SS-04-07, Stanford, California, USA, March 22–25 2004. AAAI Press.

[Qui85] Randolph Quirk. *A comprehensive grammar of the English language*. Longman Group Limited, 1985.

[Sea69] John R. Searle. *Speech Acts*. Cambridge University Press, New York, 1969.

[Sea80] John R. Searle. Minds, brains and programs. *Brain and behavioral sciences*, 3:417–424, 1980. Reprint:Haugeland, Mind Design, 1981.

[ST03] Lenhart Schubert and Matthew Tong. Extracting and evaluating general world knowledge from the brown corpus. In *Proceedings of the HLT-NAACL 2003 workshop on Text meaning*, pages 7–13, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[SVM06] Rion Snow, Lucy Vanderwende, and Arul Menezes. Effectively using syntax for recognizing false entailment. In HLT/NAACL-06 [HLT06], pages 33–40.

[tip93] Tipster complete, 1993.

[Tou03] Stephen E. Toulmin. *The Uses of Argument*. Cambridge University Press, July 2003.

[Tur50] Alan M. Turing. Computing machinery and intelligence. *MIND*, 59:433–460, 1950.

[VBA+95] Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. A model-theoretic coreference scoring scheme. In *MUC6 '95: Proc. of the 6th conf. on Message understanding*, pages 45–52, Morristown, NJ, USA, 1995. ACL.

[WB83] Y. Wilks and J. Bien. Beliefs, points of view and multiple environments. *Cognitive Science*, (8):120–146, 1983.

[WB03] René Witte and Sabine Bergler. Fuzzy Coreference Resolution for Summarization. In *Proceedings of 2003 International Symposium on Reference Resolution and Its Applications to Question Answering and Summarization (ARQAS)*, pages 43–50, Venice, Italy, June 23–24 2003. Università Ca' Foscari.

[WF94] M. Wooldridge and M. Fisher. A decision procedure for a temporal belief logic. In D. M. Gabbay and H. J. Ohlbach, editors, *Temporal Logic - Proceedings of the First Interliational Conference (LNAI Volume 827)*, pages 317–331. Springer-Verlag: Heidelberg, Germany, 1994.

[Wit] René Witte. Multi-lingual noun phrase extractor.

[Wit77] Ludwig Wittgenstein. *Philosophische Untersuchungen*. Suhrkamp, Frankfurt am Main, Germany, 1st edition, 1977.

[Wit02a] René Witte. *Architektur von Fuzzy-Informationssystemen*. BoD, 2002. ISBN 3-8311-4149-5.

[Wit02b] René Witte. Fuzzy Belief Revision. In *9th Intl. Workshop on Non-Monotonic Reasoning (NMR'02)*, pages 311–320, Toulouse, France, April 19–21 2002.

[Zad65] Lotfi A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.