**Dresden University of Technology**
**Department of Computer Science**
Institute for Systems Architecture
Chair of Computer Networks

Diploma Thesis

# NewsX
## Event Extraction from News Articles

**Author**
Martin Wunderwald
born 02.12.1982 in Dresden

**Supervisor**
Dipl.-Medien-Inf. David Urbansky

**Professor**
Prof. Dr. rer. nat. habil. Dr. h. c. Alexander Schill

**Date of submission**
March 10, 2011

# Contents

# 1 Introduction

The world is fast changing, a statement known to many people. For these people globalization intensifies the need to adapt on events happening (e.g in stock market). Therefore they must accomplish to retrieve and comprehend a huge amount of information in less time. The increasing degree to which world events are captured enhances our ability to observe global news at any time. Further, it enables us to recollect reasons about real-world occurrences, and relate to other events. However, due to the lack of structure and the heterogeneity of information sources, access to this huge collection of information has been limited to browsing, searching and reading through articles. To cope with that enormous amount of data available, Information Extraction Systems have been developed whose goal is to extract structured information from unstructured machine-readable documents.
*Information Extraction* (IE) is defined as the process of selectively structuring and combining data that are explicitly stated or implied in one or more natural language documents. There is a considerable interest in using these technologies for *Information Retrieval* (IR), since there is a increasing need to localize concise information in documents. For instance answering a given question is not only done by retrieving the entire document. Advanced retrieval models such as language modeling are building a probabilistic model on the content of a document to answer that need. Question Answering is taking a next step by inferring answers to a natural language question from a document collection. In such information retrieval models a classification of entities, relation between entities and semantically relevant parts of text is very valuable to text searching. Semantic classification becomes essential when talking about semantic Web, but also in other tasks such as text summarization and information synthesis from different documents, Information Extraction is an essential preprocessing step.

Besides extraction and identification of specified classes of names and entities, relation and event extraction is a major task of IE. Events are real-world occurrences that unfold over time and space and typically will involve certain change of state, thus many domains are characterized by key events or scenarios. Formally, the task of event extraction is to automatically identify events in free text and to derive detailed information about them. Ideally an event extraction system answers *Who did what, when, where and possibly why and how.* Due to the complexity of natural language and due to the fact that in news articles a full event description is scattered over several sentences and documents, extraction of events is a higher-level information extraction task which is not trivial. Recognizing the different forms in which an event can be expressed, distinguishing events of different types and finding the arguments of an event are all challenges.

## 1.1 Motivation

Every day hundreds of news articles are being published by the main news agencies (e.g Reuters, Aljzeera, CNN, BBC). As the amount of news articles on-line increases rapidly, so too does the need to find the right news article quickly and efficiently. The readers task is to filter out the desired information from headlines and teasers by scanning various sources of news articles. If the reader wants to get into detail he has to browse various websites and read through several articles about the same issue with facts he even already knows.

A reader not willing to read through a collection of news articles needs a representation of the news in a compact form describing the topic concisely. Catchwords are not enough, since they do not contain enough semantic information to understand an event completely. A famous concept in news information gathering is the concept of the five "W" and one "H". The 5W1H originally states, that a news story should be considered as complete if it answers the six questions, Who, What, Where, When, Why and How. The answers to this six questions are elaborate enough to understand the whole story (Carmagnola, 2008).

Given such structural characteristics of a news article we are capable of further semantic processing, that for example could lead us to build up a large scale news semantic knowledge base. On such a knowledge base we can apply news filtering and topic categorization technologies or build up a time-dependent event hierarchy. Furthermore it enables the efficient delivery of customized content related to companies, people, products and countries.

## 1.2 Use Case

In this section the use cases described show possible uses of the 5W1H extraction system.

Bob provides news articles from various news agencies in form of browse-able web content. Each news article is unstructured and embedded in a webpage. A user, Eve, consumes the provided information about a topic she is interested in by browsing through the given articles. Since Eve only needs a clear and brief overview about the topic-related event described in the article she does not want to check and verify all information contained. The only information she is interested in is who did what, when, why, where and optionally how did it happen. To gain the desired information she clicks a button and concise information in form of answers to the 5W1H questions are being extracted and displayed.

Alice is also interested in Bob's news articles and wants to build up a database filled with news. As Alice wants to make the news accessible for further semantic interpretation such as clustering by people and organizations involved in the event, she decides to break down the news articles to the 5W1H concept. So she collects the URLs of Bobs news and applies a light-weight 5W1H extraction system that stores each event into her database. Now Alice develops an application for accessing her

database as an network of related news, by the help of the previously extracted 5W1H key information. This network enables Tom, a Reporter, to gather news articles related to the topic he is writing about. In an interactive visualization he can easily see the temporal, local and contextual relation between the events through their connection. The more fine grained he specifies the request to the network, the more concrete it delivers related events. Thus he gains an overview about his topic with people, locations and events related. Alice already thinks about developing a software that builds up causal chains over events by connecting them through the extracted 5W1H semantic elements.

## 1.3 Research Questions

This thesis aims to contribute towards the automatic extraction of answers to the 5W1H questions from a news article within the web knowledge extraction system WebKnox [1]. The practical topic of this thesis is to support readers to comprehend a news article faster by giving them the answers to the questions Who, What, Where, When, Why and How.

In the course of this thesis, the following research questions will be addressed:

1. What percentage of news articles contains answers to the questions Who, What, Where, When, Why and How.
2. What are the relevant features of information and document structure in news articles, that can be helpful to automatically extract the answers to the 5W1H questions.
3. What methods from the fields of *Natural Language Processing* (NLP) and *Information Extraction* (IE) can be conductive to the extraction of the needed information.
4. How can extracted answers be conductive to the extraction of other answers.

## 1.4 Theses

The following theses The study makes the following research contributions.

1. The system should be able to extract the 5W1H semantic elements from different sources such as webpages.
2. The system should be able to extract the 5W1H from any domain of news.
3. The system applies *Named Entity Recognition* (NER) and *Co-reference Resolution* (CR) on the news article and uses a *Machine Learning* (ML) approach with derived features of the recognized entities to extract the `WHO` and `WHERE`.

[1] `http://www.webknox.com/`, accessed on 04/03/2011

4. To extract the verb describing WHAT happened, the system should apply *Parts of Speech* (POS) *Tagging* and *Phrase-Chunking* on sentences containing the WHO.

5. To extract the WHEN, the system uses features of the document structure or temporal references from the text.

6. To extract the WHY and HOW, the system uses custom regular expressions to find the relevant sentences.

## 1.5 Structure

The following chapters of this thesis are structured as described below:

- Chapter 2 introduces to Machine Learning and illuminates its tasks such as classification. We explain applications and the different statistical and symbolic approaches that have been made to solve Machine Learning problems. In Section 2.2 we describe the various techniques from the field of Natural Language Processing that can be conductive to extract the 5W1H. In the following Section 2.3 we give a brief overview of Event Extraction and will shed light on historical development of this field of research. Furthermore we investigate three different approaches of EE that intersect with different fields of research. A definition of the 5W1H task and presentation of two exemplary related works closes the chapter.

- The concept is described in Chapter 3 and introduces the system architecture as well as the extraction mechanisms of the 5W1H. The ideas how each particular extraction of the 6 answers is being solved are explained briefly.

- In Chapter 4 we give an insight in the implementation of the NewsX system. The applied techniques and used components of the previously described concept and architecture are being briefly explained. Besides introducing the extraction process, we also characterize problems that occurred while implementation stage.

- Evaluation chapter starts with evaluation of different classification algorithms and the results. The core part of this chapter is the evaluation of the NewsX system by a user study. We discuss the results and give recommendations for improvement of the system.

- The final Chapter 6 concludes the thesis with a summary of the contributions and proposes topics that should be considered in future work.

# 2 Background

## 2.1 Machine Learning

Nature has designed humans as an advanced neural- cognitive system to learn, recognize and make decisions. Recognizing a face or learning and understanding the spoken languages are astoundingly complex tasks, that are internally performed by an coordination between eye, ear, hands etc. Humans learn from experience, by mistakes or successes. By human nature it is natural that we seek to invent and build systems that can also learn to perform these learning, recognition and decision making tasks.

So in the field of *Machine Learning* (ML) one considers the question of how to make machines able to "learn". In this context things learn when they change their behavior in a way that makes them perform better in the future (Witten and Frank, 2005). More specifically, Machine Learning is a method for creating computer programs by the analysis of data. Some ML systems are based on a collaborative approach between humans and machine, while others attempt to eliminate the need for human intuition in the analysis of the data. ML is a natural outgrowth of the intersection of Computer Science and Statistics.

In the nineties there has been a profound shift in computational linguistics from manually constructing grammars and knowledge bases to partially or totally automating this process by the use of statistical learning methods that are trained on large natural language corpora. The popularity of ML algorithms has its origin in the performance improvement of hardware and software architectures that allow the huge amounts of information to be processed. Distribution of statistical learning methods was also pushed forward by the growing availability of large machine readable corpora from different sources, languages, levels of annotation, etc. that could be used by researchers to evaluate and compare their systems.

### 2.1.1 Tasks and Applications

Over the past years study of Machine Learning has spun off an industry in data mining to discover hidden regularities in the growing volumes of online data. The niche where it will be used is growing rapidly as applications grow in complexity, as the demand grows for self-customizing software, as computers gain access to more data, and as we develop increasingly effective ML algorithms.

In NLP the choice of automatically processing such massive quantities of free text has contributed to the development of various methods and techniques with an application to a huge variety of natural language problems, for instance automatic extraction of lexical knowledge, lexical and structural disambiguation, IE and IR, automatic summarization, machine translation, parsing etc.

The starting point for solving the most NLP problems is to find and create structures in unstructured data, which is the main goal of classification and clustering techniques.

#### Classification

The core of problems addressed by ML techniques, appearing at all levels of the language understanding process, are those of natural language disambiguation. These problems are particularly appropriate because they can be reformed as *classification problems*, which are a generic type of problems with a long tradition in the *Artificial Intelligence* (AI) area. Given a set of classes, classification seeks to determine which class(es) a given object belongs to. Like books in a library are being assigned to their categories by a librarian, many classification tasks have been solved manually. An alternative approach is the classification by the use of rules, that are most commonly written by hand. Such a set of rules captures a certain combination of keywords that indicates a class. Since creating and maintaing hand-coded rules over time is labor intensive, there is a third approach to text classification, namely ML based text classification. In ML the decision criterion of a classifier is learned automatically from training data. These training data have to be a number of good example documents for each class, where a person have labeled or annotated the data in each document with its class. Of course labeling data is an easier task than writing rules (Manning et al., 2008). This type of ML is called *supervised learning* because a supervisor defining the classes and labels is needed to direct the learning process.

Classification can be seen as two separate problems - binary classification and multiclass classification. Binary classification requires discerning between two classes, whereas multiclass classification involves assigning an object to one of several classes. Since many classification methods have been developed for binary classification, multiclass classification is carried out by serially applying binary classification.

#### Clustering

Coming from supervised text classification, clustering is the most common form of *unsupervised learning*. Thus, there is no human expert who has to assign objects to their classes. Clustering algorithms group a set of objects into a subset of *clusters*. The algorithms goal is to create clusters that are coherent internally, but clearly different from each other. Objects within a cluster should be as similar as possible; and objects in one cluster should be as dissimilar as possible to the objects in the other clusters (Manning et al., 2008).

### 2.1.2 Approaches

Most of the corpus-based language acquisition methods applied by NLP researchers were borrowed from statistics and information theory. As a consequence of this col-

laboration, well-known statistics based techniques could be adapted to the particular problems of NLP and lead to a significant progress in this field of research.

Learning approaches are usually categorized as statistical (also probabilistic or stochastic) methods and symbolic methods that do not explicitly use probabilities in the hypothesis.

### Stochastic Machine Learning Approaches

A stochastic model is defined as a model that describes the real world process by which the observed data are generated (Dietterich, 1997). The typical representation of a stochastic model is a probabilistic network that represents the probabilistic dependencies between random variables. Each node in the graph has a distribution, and from these individual distributions, the joint distribution of the observed data can be computed. There are different approaches that vary in how the probabilistic network is acquired and in which is the method applied to combine individual probability distributions.

### Naïve Bayes

The most simple approach to stochastic classification is to use the Naïve Bayes Classifier, which is based on the Bayes' theorem and the assumption of independence between features. Despite its simplicity it has been widely used in the ML and NLP communities with surprising success.

Naïve Bayes classifiers assume that the effect of a variable value on a given class is independent of the values of other variables. This assumption is called class conditional independence. It is made to simplify the computation and in this sense considered to be naive. Studies have exhibited high accuracy and speed when applied to large datasets.

### Maximum Entropy

Lau et al. (1993) have proposed an approach for combining statistical evidence from different sources, that is based on the *Maximum Entropy* (ME) Principle. This work was originated within the speech recognition field, but it has also been successfully applied to different NLP problems (Ratnaparkhi, 1998).

The ME approach is based on linear regression, which task is to find a linear equation for input features and predicting real-valued output. But often in speech and language processing we are doing classification, in which the output we are trying to predict takes on one from a small set of discrete values. Considering the simplest case of binary classification you can only take on the values 1 (true) or 0 (false). In such cases *logistic regression* can be used to classify an observation into one of two classes. In most of the time, the kinds of classification problems that come up in language processing involve larger numbers of classes, for instance a set of POS classes. Logistic regression can also be defined for such functions with many discrete

values, which we call *multinominal logistic regressions* (in fields of language processing also called *MaxEnt*).

Classification in MaxEnt is thus a generalization of classification in Logistic Regression. In *boolean logistic regression*, classification involves building one linear expression that separates the observations in the class from the observations not in the class. By contrast, classification in MaxEnt, involves building a separate expression for each of the classes.

Logistic regression is a function approximation algorithm that uses training data to directly estimate $P(C \mid X)$, in contrast to Naïve Bayes. In this sense, logistic regression is often referred to as a *discriminative* classifier because we can view the distribution $P(C \mid X)$ as directly discriminating the value of the target value $Y$ for any given instance $X$.

### Hidden Markov Models

*Hidden Markov Models* (HMM) had their major success in the low-level tasks of language disambiguation, that is speech recognition and synthesis, POS-Tagging, and NER. In a HMM, it is hypothesized that there is an underlying finite state machine that changes state with each input element. Each state transition is assigned a probability and generates an output sequence by some probability function (Freitag and Mccallum, 1999).

The models are called *hidden* Markov models because only the output symbols can be observed, but not the underlying state sequence. For estimating the probability of name versus non-name readings, as needed in a name tagger for instance, it is presumed that the individual events are part of some larger constituents, such as names of particular type. Whether a word is part of a name or it is not, is a random event with an estimable probability. When the recognizer is being run, it computes the maximum probability path through the hidden state model for the input name sequence, thus marking spans of input that correspond to names. The search algorithm usually used to find such a path is called the *Viterbi algorithm*. This algorithm is well explained in literature on speech recognition.

### Symbolic Machine Learning Approaches

Statistical methods in general are hindered by the data sparsity problem. Symbolic approaches exploit structural aspect of data, and use structural or symbolic information.

### Decision Trees

Decision tree based methods represent one of the most popular approaches within the AI field for dealing with classification problems. They have been used for years in several disciplines such as statistics, engineering, decision theory and signal processing. The basic idea involved in a decision tree approach is to break up complex decision into a union of several simpler decisions. Decision trees are easy to understand and

modify, and the model developed can be expressed as a set of decision rules to predict the target variable. A decision tree is a class discriminator that recursively partitions the training set until each partition consists entirely or dominantly of examples from one class. Each non-leaf node of the tree contains a split point that is a test on one or more attributes and determines how the data is partitioned. Classification of a new test point is achieved by moving from top to bottom along the branches of the tree, starting from the root node, until a terminal node is reached. Decision trees are simple yet effective classification schemes for small datasets. The computational complexity scales unfavorably with the number of dimensions of the data, thus large datasets tend to result in complicated trees, which in turn require a large memory for storage.

#### Artificial Neural Networks

Neural networks, suggested first by Turing (Ince, 1992), are a computational model inspired by the connectivity of neurons in animated nervous systems. Via the Universal Approximation Theorem by Haykin (Kubat, 1999) it was proven that neural networks can approximate any function mapping. In a neural network each circle denotes a computational element referred to as a neuron, which computes a weighted sum of its inputs, and possibly performs a nonlinear function on this sum. If certain classes of nonlinear functions are used, the function computed by the network can approximate any function, specifically a mapping from the training patterns to the training targets.

#### Support Vector Machines

Support Vector Machines (SVM) were introduced by Vapnik in 1979, but have only recently been gaining popularity in the ML community (Vapnik, 1995). They are based on the principle of *Structural Risk Minimization* from Computational Learning Theory. In their basic form, SVMs construct a maximal margin hyperplane in a high or infinite dimensional space that separates a set of positive examples from a set of negative examples with the maximum margin. Often in this space the sets of data points to be discriminated are not linearly separable. It was proposed to map the original finite dimensional space into a much higher or infinite space to make the separation easier. Transforming the space back into the lower dimension space, the linear hyperplane becomes a nonlinear hyperplane, clearly separating the data points into two classes. The separating planes are optimal, which means that a maximal margin classifier with respect to the training data set can be obtained.

An important and unique feature of this approach is that the solution is based only on those data points which are at the margin, called *support vectors*. This means SVMs automatically select their model size by selecting the support vectors, thus unlike Artificial Neural Networks, the computational complexity of SVMs does not depend on the dimensionality of the input space. Here SVMs can provide a significant improvement.

## 2.2 Natural Language Processing

In the last decade we have witnessed an ever-growing trend of utilizing NLP technologies, which go beyond the simple keyword look-up, for automatic knowledge discovery from textual data available on the Internet. In this section we will introduce the most important technologies being used in NLP. These techniques help us to extract the information scattered across the news article as they reveal important properties of sentences, phrases and words.

### 2.2.1 Tokenization and Stemming

Identifying the elementary parts of natural language such as words, punctation marks and separators from a text is essential for text processing. In NLP this process is referred to as *Tokenization*. A *token* is an instance of a sequence of characters in a particular document that are grouped as a useful semantic unit (Manning et al., 2008).

When two character sequences are not the same, like for instance searching for USA and you want to match U.S.A as well, you need a process of canonicalizing tokens so that matches occur despite superficial differences in the character sequence. This process is called *Token normalization* and is also often referred to as *term normalization*. The resulting output, grouping equal tokens, is called *term*. Some common words in a document are of little value in helping to extract important terms from a document, such as the prepositions: “to”, “by”, “of” and “at”. These words are called *stop words*. Sorting the terms occurring in a document by frequency and picking the most frequent terms is the general strategy for determining a stop word list. The process of removing these common words is called *stop word removal*.

Concerning the internal structure of words, it is useful to reduce all words with the same root or stem to a common form. This is usually done by stripping each word of its derivational and inflectional suffixes and/or prefixes. More generally it is attempted to “reverse” the inflection process by performing the inverse operation related to the basic inflection rules. These affix removal conflation techniques are often referred to as *stemming algorithms*. A very widely used de-facto standard stemming algorithm for English is the *Porter Stemmer* written by Martin Porter. He released an official implementation of the algorithm around the year 2000 and extended his work over the following years by building *Snowball*, a framework for writing stemming algorithms.

The described methods are reducing the size of indexed words significantly, thereby facilitating resulting sequence of meaningful tokens, which is fundamental for further text processing.

### 2.2.2 Sentence Splitting

Sentences are the smallest unit for expression of complete thoughts or events and they are binding interrelated information, thus they are the most important elements

of natural language for structured representation of the written content. For these reasons it is crucial for many IE approaches to recognize the boundaries of sentences in a document. The task would be trivial if the punctuation marks were not ambiguously used. A period can end a sentence but it can also denote an abbreviation or acronym, a decimal point or email address. About 47 percent of the periods in the Wall Street Journal corpus denote abbreviations (Stamatatos et al., 1999).

Sentence Splitting is also referred to as Sentence Boundary Disambiguation (SBD). The correct representation of a text as a sequence of sentences is utilized for syntactic parsing and further semantic processing.

### 2.2.3 Part-of-Speech Tagging

Parts of Speech (POS) tagging is the process whereby tokens are sequentially labeled with syntactic labels from a list of POS tags (or tag-set), thus its task is to determine the parts-of-speech for a given sentence. A simplified form of POS tagging is commonly taught to school-age children, in the identification of words as nouns, verbs, adjectives, adverbs and some more.

Parts of Speech are also known as POS, *word classes*, *morphological classes* and *lexical tags*. Their significance is the large amount of information they give about a word and its neighbors. The knowing of POS can also contribute to speech synthesis and speech recognition, since same words are spelled different depending on there POS. For example the verb discount is pronounced *disCOUNT* and the noun *DIScount*. As you can see in this example many words have multiple possible POS. In this cases the meaning of the sentence determines the correct tagging. Considering multiple POS for each word during parsing we may pass along the ambiguity. Here the approach is to select a single POS tagging prior to parsing, using various statistical methods.

Parts of Speech can be divided into two broad categories: *open class* types and *closed class* types. Closed classes have a fixed membership, for instance prepositions are a closed class, because there is a fixed set of them in english. These closed class words are often *function words* such as “it”,“of” or “and”, occur frequently, tend to be very short and having structuring uses in grammar. By contrast nouns and verbs are open classes, because new nouns and verbs are continually coined. The major open classes are nouns, verbs, adjectives and adverbs.

To give an impression about the complexity of the certain classes, we will explain nouns and verbs in detail. The syntactic class in which the most people, places and things occur is the class of nouns. But nouns also include concrete terms such as “boat” and “chair”, abstractions such as “relationship” or “bandwidth”, and verb-like terms like “pacing”. Nouns are traditionally grouped into *common nouns* and *proper nouns*, where proper nouns are for example names of entities. In English, common nouns are divided into *mass nouns* and *count nouns*. As the name tells, count nouns can be counted (“one car”, “two cars”), thus they can occur in singular and plural, whereas mass nouns cannot be counted (“salt”).

The class of verbs contains most of the words referring to actions and processes. In the English language verbs have a number of morphological forms: non-third-person-

sg (“eat”), third-person-sg (“eats”), progressive (“eating”), past participle (“eaten”), which can be recognized. The following list contains all the open word classes and their subclasses in English.

- noun (mass, count, proper, common)
- verb (non-third-person-sg, third-person-sg, progressive, past participle)
- interjection
- adjective
- adverb (directional, locative, degree, manner and temporal adverbs)

A list of important closed word classes in English with a few examples is shown below.

- prepositions: on, under, over, near, by, at, from, to, with
- determiners: a, an, the
- pronouns: she, who, I, others
- conjunctions: and, but, or, as, if, when
- auxiliary verbs: can, may, should, are
- particles: up, down, on, off, in, out, at, by
- numerals: one, two, three, first, second, third

Some well known POS Tag-sets are, the Penn Treebank with 45 word classes (Marcus et al., 1993); the tag-set for the Brown corpus (Francis and Kucera, 1979) with 87 tags and the UCREL-7 tag-set (Garside et al., 1997) with 146 tags. As one can see, the size of recent tag-sets differs, because some tag-sets for instance distinguish types of pronouns. In table A.1 an extract of the tag-set of brown corpus which consists of 87 different tags is to be seen.

Here is a result of tagging a tokenized sentence:

```
The/DT results/NNS surpassed/VBD all/DT our/PRP$ expectations/NNS ./.
```

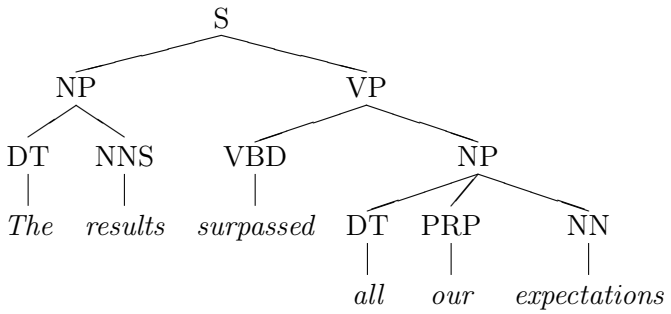
The process of *Phrase Chunking* goes a little further in showing sentence structure by separating and segmenting sentences or text strings into their subconstituents, such as noun phrases and verb phrases. Based on given POS tags it combines words to a phrase, as shown in the following example:

```
[The results]/NP [surpassed]/VP [all our expectations]/NP
```

Many algorithms have been applied to the POS tagging problem, from hand written rules (*rule-based tagging*) over statistical methods such as *HMM-tagging* and *maximum entropy tagging* up to *transformation based tagging* and *memory-based tagging*. The method used in our concept is described briefly in the concept chapter 3. Recent State-of-the-art POS-taggers reach an accuracy of 97% on the Penn Treebank Wall Street Journal Corpus.

### 2.2.4 Syntactic Processing

The next step in processing is *parsing*, where a flat input sentence is converted into a hierarchical structure that corresponds to the units of meaning in the sentence. The parser tags tokens and groups phrases into a hierarchy consisting of Parse-objects. Each possible tree for the given sentence has a probability which indicates the likelihood that this is the correct way to interpret the sentence.
There is a large variety of parsing formalisms and algorithms. Most of them consist of two main components, a grammar and a parser. The grammar is a declarative representation describing the syntactic structure of sentences in the language in a succinct way. Analyzing the input and putting out a structural representation (parse) of it, consistent with the grammar specification is task of the parser algorithm. After parsing our example sentence, we gain a structure like this, called a parse tree:




Noun Phrases (NP): “The result”, “all our expectations”
Verb Phrases (VP): “surpassed all our expectations”
Sentences (S): “The result surpassed all our expectations”

As aforementioned, CFGs serve as the nucleus of many of the parsing mechanisms. In most systems, they are complemented by some additional features that make the formalism more suitable to handle natural language.

A CFG $G = (N, \Sigma, R, S)$ where:

- $N$ is a set of non-terminal symbols
- $\Sigma$ is a set of terminal symbols
- $R$ is a set of rules of the form $X \to Y_1 Y_2 \cdots Y_n$ for $n \geq 0, X \in N, Y_i \in (N \cup \Sigma)$
- $S \in N$ is a distinguished start symbol

The problem with parsing is the ambiguity of the resulting tree, which lead to the investigation of Probabilistic Context Free Grammars (PCFG) (Booth and Thompson, 1973). In PCFGs each derivation (parse) is augmented with a probability, which is the product of the probabilities of the productions used in that derivation. In the

following example each rule is preceded by a probability that reflects the relative frequency with which the rule occurs.

- 0.4 VP → V NP
- 0.6 VP → V NP NP

Now the number of NPs expected while deriving VPs is $0.4 * 1 + 0.6 * 2 = 1.6$.

However early results of using PCFGs for parse disambiguation were somewhat disappointing. Further development lead to *lexicalized PCFGs* where head words annotate phrasal nodes. This approach drew strength from a broader interest in lexicalized grammars and was congruent with the great success of word n-gram models in speech recognition. Recent State-of-the-art parsers, for instance the Charniak & Johnson's lexicalized N-Best PCFG Parser reach an accuracy of 91.4%.

### 2.2.5 Named Entity Recognition

One of the extracted types of tokens are *named entities*. In the research community there is no clear definition of a named entity. In the 2002 CoNLL NER task the following definition is given:

> Named entities are phrases that contain the names of persons, organizations, locations, times, and quantities.

So *Named Entity Recognition* (NER) aims to locate and classify atomic elements in free text into predefined categories such as organizations, names of persons, locations, times and quantities. These categories are often concepts in a certain scenario where the named entities are instances of them. There are at least two hierarchies of named entity types that have been proposed in literature. The 2002 proposed BBN categories are used for Question Answering and consist of 29 types and 64 subtypes. [1] In 2004 an extended hierarchy by Sekine and Nobata was proposed which consists of 200 fine grained subcategories. [2]. Some recent work does not limit the possible types of entities to extract and is referred to as *open domain*.

Whereas some approaches use a simple lookup in predefined lists of for instance geographic locations and company names, others utilize statistical models that require a large amount of manually annotated training data. State-of-the-art NER systems for English reach near-human performance as the best system at MUC-7 scored 93.39% of f-measure while human annotators scored 97.67%.

Like in other NLP tasks we have the problem of ambiguity. For example the person entity "Defence Secretary Robert Gates" has an alias named "Mr. Gates" which refers to the same entity. The process of solving this problem is called Co-reference

[1] `http://www.ldc.upenn.edu/Catalog/docs/LDC2005T33/BBN-Types-Subtypes.html`, accessed on 04/03/2011

[2] `http://nlp.cs.nyu.edu/ene/`, accessed on 04/03/2011

Resolution (CR), which groups different expressions of one entity having the same referent. Besides this problem which has to be solved by an NER system, we have the problem of specifying generic entities. Where Robert Gates refers to exactly one real world occurrence, a product such as Apple's iPhone refers to multiple real world occurrences sharing the same attributes.

We now have discussed the basic techniques of NLP supporting us to extract the needed pieces of information from a news article.

## 2.3 Event Extraction

The sheer scale of online text availability has created the pressing need for automated discovering and extraction of relevant information without having to read it all. Information Extraction has a history going back at least three decades. Since retrieving and extraction of information from the Internet is one of the main challenges that came with the idea of the semantic web, various approaches have been developed to aggregate and federate unstructured (plain text) and semi-structured (e.g. tables, lists) content. Many research works, such as Snowball (Agichtein et al., 2001), Knowitall (Etzioni et al., 2004), Textrunner (Banko et al., 2007), Leila (Suchanek et al., 2006), and StatSnowball (Zhu et al., 2009) focus on triple pattern (e.g. subject-verb-object) extraction for knowledge base construction. These works confirm the application of technologies such as pattern-matching, light natural language parsing and feature-based ML for large scale practical systems.

Research in the field of EE has been an active area for the past ten years. Many event extraction systems have been reported, for example systems capable of extracting disease outbreaks (Grishman et al., 2002) and conflict events (King and Lowe, 2003; Atkinson et al., 2008; Tanev et al., 2008). In 1998 the National Institute of Standards and Technology (NIST) sponsored the Topic Detection and Tracking (TDT) project, which invest and investigated the development of technologies that could detect events in news streams, and track the progression of these events over time (Yang et al., 1998; James et al., 1998; Yang et al., 1999b). Although the project ended in 2004, event detection and extraction research was pushed forward by the *Automated Content Extraction* (ACE) program and by the DARPA-initiated *Message Understanding Conference* (MUC), and in domains outside of news such as Biomedical Text Processing (Murff et al., 2003; Ohta et al., 2006).

Traditionally, information extraction is associated with template based extraction of event information from natural language text, which was a popular task of the MUC in the late eighties and nineties (Sundheim, 1992). These information extraction tasks started from a predefined set of templates, each containing a specific subject domain and used relatively straightforward pattern matching techniques to fill out these templates with certain instances of these events. Grammar or rules (e.g regular expression patterns) were mapped on the text in order to identify the information.

In 2004 the *eVent Detection and Recognition* (VDR) Task was introduced by the ACE initiative. In this task different event types were considered: Movement,

Conflict, Business, Personal etc. The task was to detect and select the specific events and to merge them in to a given representation. We describe the Pattern-based Approach in Chapter 2.3.1. Driven by the ACE VDR task, Heng Ji proposed a serial of schemes on event-coreference resolution (Chen and Ji, 2009), cross-document (Ji and Grishman, 2008a; Ji et al., 2009), and cross-lingual (Ji, 2009), event extraction and tracking. These schemes obtained encouraging results. In Ahn (2006), David Ahn decomposed the VDR task into a series of machine learning sub-tasks (detection of event anchors, assignment of an array of attributes, identification of arguments and assignment of roles, and determination of event coreference). Results show that anchor and arguments identification has the greatest impact on ACE value.

Since most IE techniques focus on the document instead of the event itself they fail to provide semantic information for event understanding. In order to get more semantic information for the event, some event-oriented techniques have been proposed which we describe in chapter 2.3.2.

Naughton et al. (2008) investigated sentence-level statistical techniques for event classification. The result reveals that SVMs consistently outperform the Language Model (LM) technique. An important discovery is that a manual trigger-based classification approach (using WordNet to manually create a list of terms that are synonyms or hyponyms of each event type) is very robust and outperforms the SVMs on three of six event types.

Although many works on IE have been published, research has not much payed attention to evaluate the contribution of syntactic and semantic analysis using NLP techniques. *Semantic Role Labeling* (SRL) as another example of multi-way semantic relation extraction aims to derive semantic information from text supported by a lexical resource such as FrameNet (Baker et al., 1998) and PropBank (Kingsbury and Palmer, 2004). The approach of sentence-based Event Extraction is described briefly in chapter 2.3.3.

### 2.3.1 Pattern-based Approach

The work most commonly referred to as event detection is that originating from the *Topic Detection and Tracking* (TDT) research effort sponsored by DARPA. An important contribution of that research program is the recognition of the distinction between an event and a topic. As Yang (Yang et al., 1999a) note, "The USAir-427 crash is an event but not a topic, and airplane accidents is a topic but not an event".

Within Message Understanding Conferences MUC there is a *Scenario Template* (ST) task which main goal is to "extract prespecified event information and relate the event information to particular organization, person, or artifact entities involved in the event." (Marsh and Perzanowski, 1998). Systems participating in this task use information extracted and inferred from a text to fill in appropriate fields in predefined templates corresponding to the domain of the text. Since the domain is given, the semantics for the domain of interest is known and systems can achieve high performance (up to 50%-60% recall and precision in MUC-3 to MUC-7 ST tasks) on a text understanding task. MUC systems suffer from two drawbacks. First, the fixed

templates preclude detecting events of types not anticipated during system design, or multiple events of different types. Second, they are dependent on the domain, which requires a lot of time to create accurate templates defining possible events for that particular domain.

The ACE program defines a wider spectrum of event types and introduced more complex template structures. One challenge presented by the ACE program is the ACE VDR Task which aims to recognize events mentioned in text and merge them into a given representation for each detected event. All mentions of a given event are aggregated into one representation object. An ACE event is an event involving zero or more ACE entities, values and time expressions. The required output for one event includes information about the attributes of the event, the event arguments, and the event mentions. Event attributes are the event *type, subtype, modality, polarity, genericity* and *tense*. The event types and subtypes for the 2005 VDR task are listed in A.2. Each event type consists of arguments which are identified by a unique *ID* and a *role*. Unlike ACE relations, events allow multiple arguments in the same role.

The event patterns can be found in the Event guidelines and are described with their roles, explanations and examples.[3] For instance the `ACE INJURE` Events have three participant slots (`AGENT-ARG`, `VICTIM-ARG` and `INSTRUMENT-ARG`) and two attribute slots (`TIME-ARG` and `PLACE-ARG`). Each Slot is filled with one or more certain ACE entity types. In the example pattern the `AGENT-ARG` can be a *person* (PER), *organization* (ORG) or *geopolitical entity* (GPE), whereas the `VICTIM` only can be a person and so on.

At the ACE 2007 there was only one participant on the VDR task, that was BBN Technologies, who had an overall score of 13.4 with a possible maximum of 100, where at the ACE 2005 four teams participated with a maximum score of 14.4. [4]

### 2.3.2 Event-oriented Approach

Many of the occurring events will be reported multiple times, in different forms, both within the same document and within topically related documents. Event-oriented systems take advantage of these alternate descriptions to improve consistency of the extraction.

Most event-based summarization approaches rely on statistical features derived from multiple documents. Based on these features various clustering approaches have been investigated in document summarization, which differs in the kind of features they use.

Ji and Grishman (2008b) demonstrate that appreciable improvements are possible over the variety of event types in the ACE evaluation through the use of cross-sentence and cross-document evidence. In their paper the central idea of inference is to obtain document-wide and cluster-wide statistics about the frequency with which triggers and arguments are associated with particular types of events, and then use this information to correct event and argument identification and classification.

[3] `http://projects.ldc.upenn.edu/ace/data/`, accessed on 04/03/2011

[4] `http://www.itl.nist.gov/iad/mig/tests/ace/`, accessed on 04/03/2011

Based on the resulting inference rules they remove triggers and arguments with low (local or cluster-wide) confidence and adjust trigger and argument identification and classification to achieve consistency. They share the view of using inference to improve event extraction with recent research of Yangarber (Yangarber, 2006; Yangarber and Jokipii, 2005). Recently Grisham and Liao improved trigger event classification and argument role classification in ACE event extraction through document-level cross-event inference (Liao and Grishman, 2010).

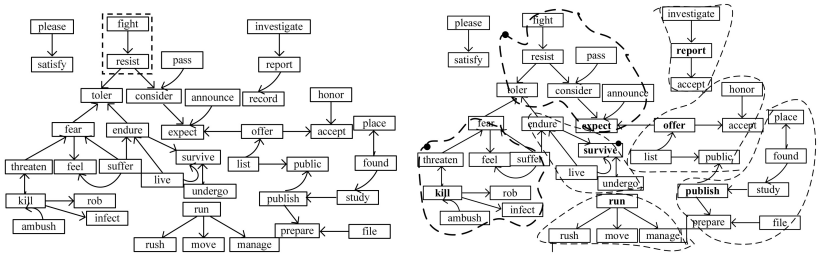
To gain more semantic information about the event, techniques such as Event-Based summarization have been proposed (Liu et al., 2007; Filatova and Hatzivassiloglou, 2004; Li et al., 2006), which extract and organize summary sentences in terms of the events that the sentences describe.

Naughton et al. (2008) focus on merging descriptions of news events from multiple sources. The first step in their approach is to identify the spans of text in an article corresponding to the various events that it mentions. Then, they identify event descriptions from different articles that refer to the same event. As a baseline for this clustering process they use the Agglomerative Hierarchical Clustering (AHC) algorithm (Manning and Schütze, 1999), which they extend by the feature of sentence position in text. The procedure ends up in a conversion of the event description into a structured form so that they can be merged in a coherent summary.

Filatova and Hatzivassiloglou (2004) apply clustering method to organize similar paragraphs into tight clusters based on primitive or composite features. They ignore sentences that do not contain at least two named entities or frequent nouns. To form the summary one paragraph per cluster is selected. (Zha, 2002) used spectral graph clustering algorithm to partion sentences into topical groups. Within each cluster, the saliency scores of terms and sentences are calculated using mutual reinforcement principal, which assigns high salience scores to the sentences that contain many terms with high salience scores. To generate the summary, the sentences and key phrases are selected by their saliency scores.

Since the granularity of clustering units mentioned above is rather coarse, Liu et al. (2007) defined a event term as clustering unit and implemented a clustering algorithm based on semantic relations. They approximately define the verbs and action nouns as the event terms which characterize the event occurrence. Then they extract these event terms from the documents and construct the event term graph by linking terms with the semantic relations derived from an external resource as one can see in 2.1a. Finally they group the similar and related event terms into the cluster of the topic 2.1b.

Many different variations of this extractive approach (Jones, 2007) have been tried in the last decade. However, it is hard to say how much greater interpretive sophistication, at sentence or text level, contributes to performance. Without the use of NLP, the generated summary may suffer from lack of cohesion and semantics. If texts containing multiple topics, the generated summary might not be balanced. Deciding proper weights of individual features is very important as quality of final




(a) Terms connected by semantic relations

(b) Generated Term clusters

Figure 2.1: Term Clustering Process in (Liu et al., 2007)

summary is depending on it. Exemplar systems are NIS (News in Essence) [5] which is based on MEAD (Radev et al., 2004), the Columbia Newsblaster (McKeown et al., 2003) and GISTexter (Finley and Harabagiu, 2002).

### 2.3.3 Sentence-based Approach

Recently, *Semantic Role Labeling* (SRL) aiming to derive detailed semantic information from a sentence is utilized in Question Answering Systems to extract the 5W1H from a sentence (Yaman et al., 2009).

Different from full semantic parsing, SRL only labels semantic roles of constituents that have a direct relationship with the predicates (verbs) in a sentence. Typical semantic roles include agent, patient, source, goal, and so on, which are core to a predicate, as well as location, time, manner, cause, and so on, which are peripheral. Such semantic information can be important in answering 5W1H of a news event. Surdeanu et al. (2003) designed a domain-independent IE paradigm, which filled event template slots with predicate and their arguments identified automatically by a SRL parser as one can see in Figure 2.2.

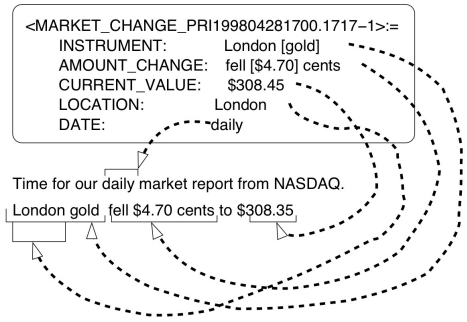



Figure 2.2: Templette filled with information about a market change event

[5] http://newsinessence.com/, accessed on 04/03/2011

In 2006 a workshop on event extraction and synthesis in conjunction with AAAI-06 took place. One focus of the workshop was the role of semantics in event extraction. As part of the workshop, combination of statistical event extraction using Machine Learning over PropBank semantic role labels and finally mapping entity relations to domain ontology was reported as a promising approach (McCracken, 2006).

Fillmore et al. (2006) show that the kind of information produced by the lexicon-building project FrameNet can have a special role in contribution to text understanding. The FrameNet project is devoted to discovering and describing the semantics and syntactic combinatorial properties of lexical units in English, and how these properties can be used for identifying and populating the eventualities that are linguistically coded in a document. The most straightforward way eventualities can be filled in is through the recognition of frame-bearing words that designate eventualities of particular types and identification of such words that denote participants ("slot fillers") in these eventualities. However, semantic parsing is still computationally intractable for larger corpus.

### 2.3.4 The 5W1H Task

As one can see, there are various approaches aiming on event extraction having their origin in different fields of research, from statistical techniques over document clusters to sentence-level semantic analysis. All these event extraction approaches have in common that they want to extract a detailed description of an event. Since their is no consistent form of an event description, ACE used event patterns referring to certain event types as templates to be filled by the system. The advantage of using those patterns, is that the system is easy to evaluate on the other hand we have the problem that not each event fits in one of these patterns. Thus we have to find a more domain independent representation of an event. A key maxim in journalism is to use the six interrogatives - *who?*, *when?*, *where?*, *what?*, *why?*, and *how?* to develop a comprehensive reportage of the event. These event facets, referred to as 5W1H for short, originally state that a news story should be considered as complete if it answers a checklist of these six questions. The factual answers to these questions are elaborate enough for people to understand the story as a whole.

Recently the 5W1H concept is utilized in sentence-level understanding tasks, where it seeks to summarize the information in a natural language sentence by distilling it into the answers to the 5W questions (Parton et al., 2009). In these tasks the 5W refer to semantic roles within a sentence. More news event related applications of the 5W1H concept can be found in Xie et al. (2008), where they review new technologies and systems identifying and analyzing events and activities in multimedia streams. We adopt the six facets to describe events in our event extraction system, because they are key semantic attributes that are sufficient and necessary to summarize an event, as prescribed by journalism principles.

## 2.4 Related Work

In the related work part we explain two systems which differ in their approaches but have the common goal to extract 5W1H from a news article.

### 2.4.1 NEXUS

News cluster Event eXtraction Using language Structures (NEXUS) is and event extraction system utilized for populating violent incident knowledge bases. It automatically extracts security-related facts from online news articles. Before the NEXUS event extraction process can proceed, news articles are gathered by media monitoring software (EMM system), which delivers news clusters for each topic. Further, NEXUS selects security related events via application of key-word based heuristics. In the next step the documents in each cluster are linguistically preprocessed which encompasses the following steps: Sentence Boundary Disambiguation, Named Entity Recognition, simple chunking, labeling of action words (e.g. kill, shoot) and unnamed person groups (e.g. six civilians).

In the subsequent step, the pattern engine applies a set of hand-coded extraction rules on each document within a cluster and core templates are learned from annotated data. Based on the resulting templates they apply a bootstrapping ML approach on the whole cluster, which guarantees better precision of the learning patterns. They extract anchor entities from the news corpus and learn additional patterns from the context of these anchor entities. The following example will explain this in detail.

A learned core template `killed X` is matching a text like "U.S. troops killed Al Zarkawi". *Al Zarkawi* is taken as an anchor entity and consider the contexts in which this name appears is the same news cluster where the above mentioned text appears. For example: "the body of Al Zarkawi was identified", "the body of Al Zarkawi was found" etc. All these contexts from different anchors are passed to a pattern learning algorithm to extract candidate patterns (e.g. `the body of X`). The criterion for the pattern selection is based on the so called *Local Maximum of the Context Entropy*, which they define as follows:

> We have Local Maximum of the Context Entropy in the pattern $P$ when all the patterns which immediately precede it have the same or lower context entropy and all the patterns which p precedes immediately have lower context entropy.

Finally, they select only the patterns in which they have a local maximum of the context entropy. The set of pattern is expanded by synonyms and hyponyms from WordNet and by syntactic variants. Due to known data sparsity problem they manually add hard-to-learn patterns to the library. In total they learned 3415 templates for `affected_dead`, `affected_wounded`, `affected_kidnapped` and `perpetrators`. In order to fill slots which are certain semantic categories like for instance *weapons*, they perform a acquisition of a lexicon using the hypernym chain from WordNet. To

fill the event frame describing the main event they perform an event aggregation containing the following steps:

- victims: resolving role ambiguities of recognized entities
- number of killed, wounded and kidnapped: average-like estimation
- place: geocoding via EMM system
- perpetrators: named entity recognition
- weapons: lexicon
- event type: classification over lexicon of event keywords
- date: date of the news cluster

Regarding the identification of places they locate the country with accuracy of 95% but detect city, town and village with only 28%. The date could be recognized with a precision of 76%. For the slot `actors` the recall was measured with 63% (Piskorski et al., 2007).

### 2.4.2 Chinese News Fact Extractor (CNFE)

Another novel news event semantic extracting approach addressing the 5W1H based on one document was implemented in the CNFE. Since applying SRL for Event Extraction is computational intractable over large scale news corpora, they propose a "light" but effective method to extract the 5W1H. Their extraction pipeline includes three steps: topic sentences extraction, event classification and 5W elements extraction.

First, they identify informative sentences which contain the main event's key semantic information in the news article. Second, they combine a rule-based method (verb-driven) and a supervised machine-learning method (SVM) to extract events from these topic sentences. Finally, they recognize 5Ws with the help of specific event templates as well as event trigger's valency and syntactic-semantic rules. They treat the topic sentences, actually as short summarization of the news as "How" of the event and replace "Why" with "Whom" currently. Thus, they obtain a tuple of 5W `<Time, Location, Subject, Predicate, Object>` as "How". 2.3 shows the framework of the CNFE. As one can see the event type identification module accepts three inputs: topic sentences with word segmentation and POS tags, a trigger-event-type/subtype table and a set of syntactic-semantic rules of triggers. The list of event types/subtypes and associated triggers are extracted from ACE 05 training dataset.

Extraction of the 5W1H is based on the extracted topic sentences. The type identification module searches the topic sentences by examining trigger list and marks each appearance of a trigger as a candidate event. From the output of event type identification and SVM Rectifier they get headline, topic sentences and a list of 5W candidates of an event, i.e. predicate, event type, named entities, time and location words. Next is to identify 5W1H semantic elements as follows:

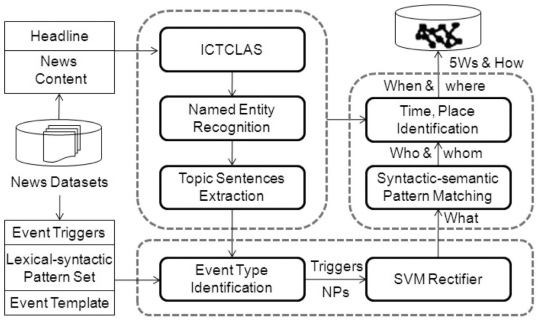



Figure 2.3: The framework of the Chinese News Fact Extractor

- **Who, Whom:** To identify these arguments, regular expressions are used to match trigger's syntactic-semantic rules. For example, they use an expression "(.*)/n(.*)/trigger(.*?)/n(.*)/n.*" to match "NP1+V+NP2+NP3". They identify arguments from Named Entities and NPs of the trigger according to the sentence's syntactic structures. Then they determine there roles (e.g. agent, patient) and associate them with a specific template.
- **What:** They use the first identified Verb of their verb-driven method which is rectified by a SVM.
- **Where, When:** Outputs of the NER are used to identify time and location. If there are no Time/Location Named Entities, generated chunks with tags of /nt and /ns are adopted.
- **How:** They combine the results to a sentence "Who did What to Whom".

They replace the Why with Whom, which enables them to rely on their topic sentences, because the answer to the question Why is scattered over the document and cannot be answered by the topic sentence. So they suppressed an important part of the original 5W1H concept, which is relevant regarding further semantic processing and setting different news into relation. On the other hand removing the why, enables them to keep their research consistent with ACE event extraction in order to compare with other works.

They evaluate CNFE on a real world corpus containing more than 30000 newspaper documents to extract ACE events, based on (Surdeanu et al., 2003) predicate-argument evaluation.

Their system is error prone due complexity of chinese language (compound sentences, special syntactic structures). Wrong segmentation and POS tags, for example, a trigger is segmented into two words and a verb trigger is wrongly tagged as a noun, have a strong impact on the result. The main problems of their method which causes wrong assignment of arguments, lie in absence of Co-reference Resolution and wrongly identified NPs (Wang et al., 2010).

# 3 Event 5W1H Elements Extraction

In this chapter we describe the concept of our system derived from the research questions and theses from chapter 1.3. We discuss problems occurred in conceptional stage and present our solution statements.

## 3.1 System Architecture

This section introduces the coarse architecture of our event extraction system. To begin we give an overview of the components, their placement within the system and their tasks. A more precise description of how each question of the 5W1H is being answered is explained in the subsequent sections.

Our system basically relies on tools from natural language processing described in chapter 2.2. The input of our system is a document with headline and text from any possible source. We implemented a gradual detection of the 5W1H with a resulting processing chain that consists of various interacting processing steps. Starting with detection of named entities, performing co-reference resolution and finally running two classifiers, to extract the `WHO` and `WHERE`, we proceed with more fine grained natural language processing on sentence level to extract the `WHAT`, where we use the results of the extracted `WHO`.

In general, we pick up the principle of deferred commitment from Yangarber Yangarber (2006), which says that each non-immediate reference should be linked to a distribution of answers in form of a ranked list of value-confidence pairs. As a result we are able to refer to these lists and can change values according to downstream processing steps. This is very useful, since subject and verb are closely coupled in a sentence and thus the who and what in our system.

Independent of the `WHO`, `WHERE` and `WHAT` extraction, there are three further chains, one for each left part of the 5W1H: `WHEN`, `WHY` and `HOW`. The extraction mechanisms for them bears on sentence detection and pattern matching.

Looking at the system as a whole, we are aiming to extract every part of the 5W1H, but not necessarily every slot can be filled, since for instance the answer to how something happened might not be given in the text. In the following sections we describe the extraction mechanisms briefly.

## 3.2 Who

We investigated several news articles and pointed out that the answer to the question "Who?" is the carrier of the statement. Thinking about "Who did what", suggests to answer the question with a certain person, a concrete named entity. But not only

people can be identified as Who, but also groups of people (e.g. victims, protestors), organizations, countries. Sometimes a WHO is not obvious, like in natural catastrophes such as volcano eruption or flood, but it is the subject of the article. So basically the WHO acts as the subject of the news article.

The assumption we made about the subject of an article is that it occurs frequently in the whole article and most likely is a named entity, so we decided to apply Named Entity Recognition on the whole article including the headline. The resulting lists of entities we extend with noun phrases from the headline, because we found out that the noun in headline in 80% of the 100 cases we investigated is our subject, but not always is a named entity. This also matches with the linguistic and structural features of a news story, that is, trying to attract readers' eyeballs with a headline (Dorr et al., 2003).

In the following example we describe the process of extracting and grouping of similar entities. 3.1 shows an excerpt of an article with annotated entities.

> [Clinton] and [Gates] visit [Korean] Demilitarized Zone. The [US] Secretary of State, [Hillary Clinton], and Defence Secretary [Robert Gates] have visited the Demilitarized Zone separating North and [South Korea]. [Mr Gates] said they wanted to show solidarity with their allies in [Seoul].

Figure 3.1: Excerpt from a text with annotated entities

After extracting a list of chunks, annotating entities and noun phrases, we group similar chunks via Co-reference Resolution. The resulting structure now holds a set of distinct entities and noun phrases linked to their occurrences within the text.

- Clinton, Hillary Clinton
- Gates, Robert Gates, Mr Gates
- Korean, South Korea
- Seoul
- US

For each group of entities we calculate the features one can see in Figure 3.2 to apply a Machine Learning (ML) algorithm.

**The number of occurrences in text** represents how often a certain entity is mentioned in the text. We assume that the number of occurrences of an entity is strongly connected to its importance.

**The number of occurrences in the headline** in the most cases is 1 or 0, which means the entity is mentioned in the headline or not. Since the headline acts as an eye catcher as described in section 3.2 it most likely contains relevant information such as the `WHO` or `WHAT` which helps to classify the entity correctly.

**Distribution** means the position of the entities within the text. We calculate it as the average position of the entity relative to the length of the text. Since the headline and sentences are at the beginning of the text, we assume that the most important entities are at the beginning of the text as well.

**Type of the entity** depends on the used Named Entity Recognizer. Since we i focusing on locations for our `WHERE` and on people, organizations and locations for the `WHO` we do not need a more fine grained distinction between the entity types. We assume that the type of the entity as feature has an high impact on the classification.

Figure 3.2: Features for the Who Classifier

We investigated four different classification algorithms: Bayes Network, Decision Tree (J48), Bootstrapping Efron and Tibshirani (1993) Bagging Breiman (1996) and Naïve Bayes. On our training set of 1000 `WHO` Candidates the Bagging Classifier performs best. In table A.4 one can see the results compared and in Chapter 5.1 the evaluation of the both classifiers is described in detail.

When the features are calculated we use our trained Bagging Classifier to classify each annotated entity with its references. The resulting list of possible `WHO` candidates is stored for further processing. At this point the highest ranked `WHO` candidate is our most likely `WHO`, but might be changed during extracting the `what` as we describe in the following section.

## 3.3 What

Referring the origin of the 5W1H, which is the Sentence “Who did What When Why and How?” and taking into consideration that our `WHAT` describes a change of state, we decide to focus on a verb as `WHAT`. In a predication which describes an event or an action caused by a verb, arguments (participants of the event) play different roles. The following example illustrates how important it is to find a `WHAT`, matching the

WHO.

Argentina's former president Nestor Kirchner has been buried.
and
Argentina buries former President Nestor Kirchner.

Figure 3.3: Example of the passive-active problem

These two headlines are reporting the same fact. The only difference is, that in the first sentence a passive voice is used whereas in the second sentence the author used active voice. It is obvious that in the first example "has been buried" is our verb phrase representing the WHAT. In case our System detects "Argentina" as the WHO, we end up with "Argentina has been buried" which is not our desired result. So the only rule we can stick to, is that word groups stand together when they belong together.

To accomplish this issue, our system performs as follows. If one of our WHO candidates occurs in the title, we look for the subsequent verb phrase of it. Now we set the WHO and WHAT as fixed for the event. If there is no WHO in the headline, we search within the text for the first occurrence of our highest ranked WHO and also take the subsequent verb phrase as WHAT. In Figure 3.4 a graphical overview is given, how the WHO, WHERE and WHAT are being extracted.

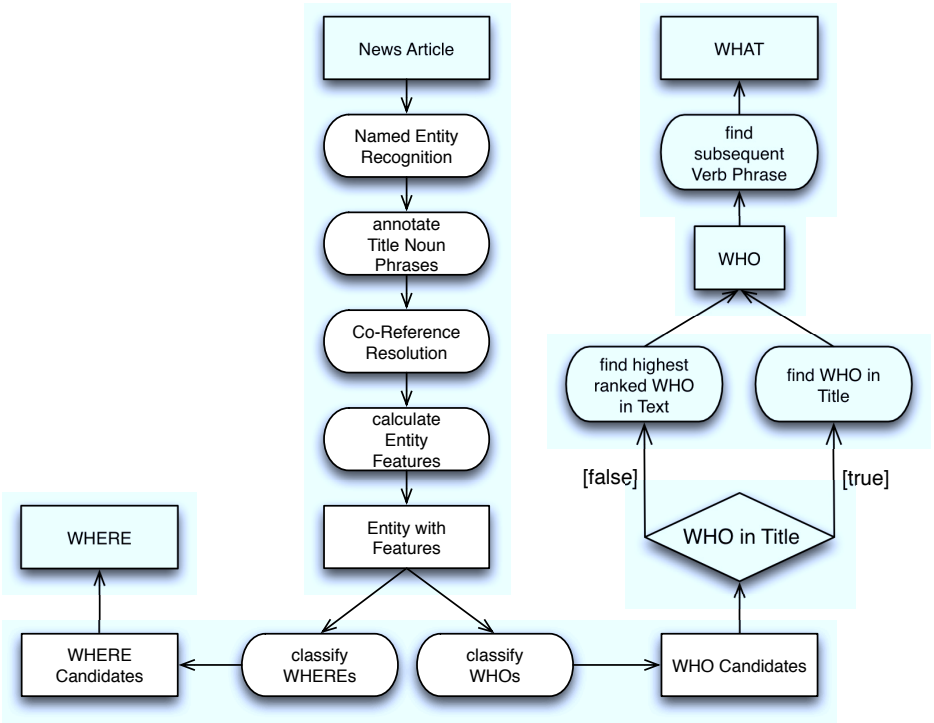



Figure 3.4: Who, Where and What processing chain

Since the highest ranked WHO most likely has multiple occurrences within the text, that are often slightly different, we use the shortest string that they have in common and search for its first occurrence. On the other hand, the shortest common string does not contain all information, thus our final output for the WHO is the longest string of the occurrences.

From this point on we do have a fixed WHO and WHAT.

## 3.4 Where

Answering the question where something happens, strongly depends on the kind of event. If it is a concrete event taking place in a certain place like when Hillary Clinton visits North Korea, the information is mentioned in the text several times or even is mentioned in the headline. But there are also abstract events, for instance when "the internet search giant Google has detailed plans to limit the number of online newspaper articles its users can read for free". Thus the where slot not always can be filled.

To extract the WHERE we make use of Named Entity Recognition because the place where something happens is most likely a named entity of type location which can be recognized through the NER. Furthermore we use the already annotated data from the detection of the WHO and thus save processing time. The difference to the WHO extraction process is, that from our annotated groups of entities we remove all entities that are no locations since only entities of type location are considered to be relevant. As a result we can eliminate the feature entity type from our WHO classifier. Finally the WHERE classifier only has to classify different locations within one article. The extracted entity with the highest confidence fills our WHERE slot.

## 3.5 When

Events are always associated with time periods or at least dates. News articles contain many temporal references for both placing the occurrences and relating them with other events. Using this temporal information can be helpful in retrieving related documents. Since we are focusing on a light weight extraction mechanism and extraction of time periods is a non-trivial task, we only extract dates that are clearly stated within the document. If there is no concrete date given within the text, we deal with the publication date of the documents, which can be used for further aggregation of related articles.

Our approach to find date expressions within the text is a simple pattern matching against time and date patterns. If no date is given and the article has an online source, we extract the publication date of the online document via a date recognition mechanism that is already part of Palladian (Gregor, 2010).

We again pick up the idea of deferred commitment and used a named entity recognizer capable of extracting time and date entities. All the extracted dates are

linked with a manually given confidence depending on the extraction method and finally saved within the event object for further processing.

## 3.6 Why

When it comes to the question `WHY`, the relational character of events becomes visible. Each event has its own internal structure, and meanwhile often relates to other events semantically, temporally, spatially, causally, or conditionally. In Section 2.3.2 we discuss event related summarization through clustering methods, where distinction between related and atomic events is essential, since only independent news articles should be grouped (Li et al., 2006).

In related events or sub-events the information, indicating a causal relation are expressed in the text in various ways. Two common ways are using *causal links* and *causative verbs*. Causal links are words used to link clauses or phrases, indicating a causal relation between them.

Altenberg (1984) provided a comprehensive typology of causal links. He classified them into four main types: the *adverbial link* (e.g. hence, therefore), the *prepositional link* (e.g. because of, on account of), *subordination* (e.g. because, as, since, for, so) and the *clause-integrated line* (e.g. that's why, the result was).

As one can see, the complexity of expression of causal relations is as high as the complexity of natural language. Furthermore it needs semantic interpretation to decide whether a causal linking refers to an event or only to the sentence within it occurs. Accurate extraction of causal relation only works on the layer of semantic interpretation or over statistical evaluation. Further reading how to acquire causal knowledge from texts can be found in (Inui et al., 2005).

We investigated our 100 news articles to find out how the words of Altenberg are indicating a causal link to another event and at the same time are the reason of the event, which answers our question `WHY`. The outcome is a list of indicators and the number each indicator is located within the sentence describing the reason of the event. For each regular expression we had a counter that counts how often a regular expression points the sentence with the reason for the event in it. If a word occurs within a different sentence, thus not indicating the reason, we decrease the counter. The resulting list contained the relation between positive and negative occurrences for each regular expression. From this list we derived confidences for each indicator which we implemented as a set of regular expressions with belonging confidences that are applied on the POS-tagged text. The sentence with the highest confidence is considered to describe the reason of our event.

One of the important observations is, that the word "since" in fact indicates relations to other events or describes relevant time periods, which can be conductive to further semantic processing, but the word in the most cases does not point to the sentences indicating the reason of the event.

The clearest indication of a reason for an event is given by the pattern where we

look for a "to verb" following the what. It is rare, but when it can be found in the text, it points to the right sentence.

Table A.3 shows the regular expressions and their confidences.

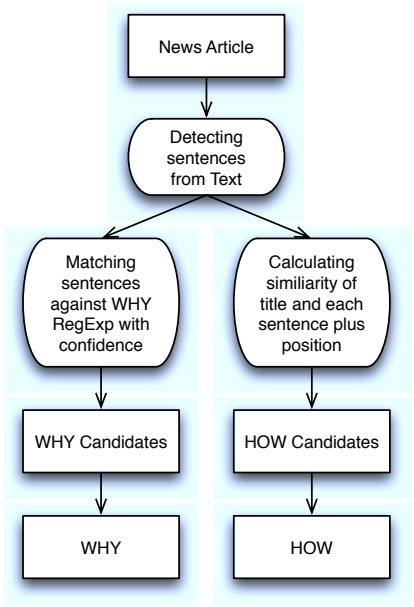



Figure 3.5: Why and Who processing chain

Finally the extraction of the WHY is the most complicated, since there are many different ways to express a causal relation which to discover mostly requires semantic interpretation. The other non-trivial task is to ensure the association of the text excerpt with the topic the news article describes, which means to point out the reason of the issue the article deals with. Despite these problems, our extraction of causal links or sentences is a key feature of our extraction system, because it opens the door to find temporally or causally related events.

## 3.7 How

How something happens is answered through a detailed description of the event itself. It depends on the needed depth of information how the answer to this question is limited. In our approach we focus on the sentence giving additional information to the title, which might be a teaser or sentence close to the title. This leads us to the assumption that the sentence that is highly similar with the extracted WHAT and WHO most likely describes our event in detail and answers the question HOW. To find this sentence, we calculate the the similarity of each sentence with a concatenated string

of WHO and WHAT. This similarity is calculated as follows:

$$similarity = \frac{longestCommonStringLength}{\min(string1.length(), string2.length())} \tag{3.1}$$

Since there are multiple sentences being similar to the title, we have multiple candidates that might answer our question, but we discovered that the position of the sentence within the text has an influence on the confidence of our answer. The sentences are positioned on begin of the article are more likely answering HOW than sentences on bottom of the text. In Section 2.4.2 they call these sentences *"topic sentences"*. Thus we increased our calculated similarity with the position index of the sentence, which leads to a ordered list of sentences with confidences. We rank the sentences by their calculated confidence and pick the first sentence as our HOW. The extraction process is visualized in Figure 3.5.

# 4 Implementation

This chapter describes the implementation of our system NewsX. It explains the implementation of the concept from the previous chapter and characterizes the components being used.

## 4.1 Palladian

The implementation of NewsX is part of the TUDIIR Palladian (Urbansky et al., 2010), which has its origin in the WebKnox project. The project was initiated in 2008 and was developed further by several student theses at Technical University of Dresden. Palladian offers functionality from the field of information retrieval such as classification, extraction of various content and crawling mechanisms. Our system NewsX extends Palladian with set of components and provides new functionality from the field of Natural Language Processing.
Palladian is implemented in Java 1.6, which is one of the most distributed programming languages and offers a collection of build-in components and libraries. Besides the build-in components there are various external libraries for nearly every purpose. Furthermore it is supported by many integrated development environments and is platform independent. Java is suited for applications in the web and can run on application servers such as Apache's Tomcat[1].

Palladian uses Apache's Maven build manager that aims to make the build process easy by providing a uniform build system. Maven is able to manage the dependencies between projects and packages. Libraries are provided through private and external repositories and can be loaded on demand.

## 4.2 Used Libraries

As aforementioned there are various libraries freely available implemented in Java. NewsX uses different external java-frameworks and libraries especially from the field of natural language processing that we characterize in this section.

**Open NLP** hosts a variety of java-based NLP tools which perform sentence detection, tokenization, POS-tagging, chunking and parsing, named entity detection and Co-reference Resolution using the OpenNLP Maxent Machine Learning package.

**LingPipe** is a free, efficient, scalable, reusable and robust tool kit for processing text using computational linguistics [2]. There is plenty of documentation available

[1] `http://tomcat.apache.org`, accessed on 04/03/2011
[2] `http://alias-i.com/lingpipe/`, accessed on 04/03/2011

on the website of alias-i. Also there can be found several tutorials for every subtask. The libraries are used in quite a number of commercial, academic, and government institutions.

**WEKA** is a open source software collection of Machine Learning algorithms for data mining tasks from the University of Waikato[3]. It contains tools for data pre-processing, classification, regression, clustering, association rules and visualization. It is also well-suited for developing new Machine Learning schemes.

## 4.3 NewsX

NewsX uses and extends the functionality of Palladian in different ways. Besides the capsuled event extraction process NewsX adds NLP functionality, such as a POS-Tagger, SentenceDetector and Parser that are concrete implementations of NLP techniques from different libraries as described in 4.2. So the integrated NLP tools can easily be reused by other components of Palladian. Basically our system consists of three parts, where the first part is the `EventProcessor`, which inherits the functionality of the `NaturalLanguageProcessor` that holds tools like POS-Tagger, Named Entity Recognizer and Sentence Detector. Furthermore, it performs the Co-reference Resolution and calculates the features of annotated entities by the help of the `EventFeatureExtractor`. The second part is the Event Extractor itself, which implements the logic by controlling the flow of data and finally constructing a complete event object consisting of the text, headline, annotated entities and the answers to the 5W1H. The sequence diagram of the described process is to be seen in Figure 4.1.

The following sections give detailed information about how the extraction process works, shows problems and explains our solutions.

## 4.4 Text Preprocessing

To extract the answers from our news article we have to apply various techniques from the field of NLP. We derive structural and contextual information such as sentence boundaries and named entities from the text and title of the news article. The derived information and event structure are held in an Event object, which also contains the ranked candidate and final answers to the 5W1H questions. Holding those information persistent enables us to reuse them during the whole extraction process.

In our investigations of the different NLP toolkits and frameworks we found out, that they offer the same basic functionality such as Parsing, Phrase Chunking, POS-tagging, Sentence Boundary Detection and Named Entity Recognition. Because of this fact we implemented abstract classes to unify these functionality. We decided to develop concrete implementations of LingPipes Phrase Chunker, SentenceDetector,

[3] `http://www.cs.waikato.ac.nz/ml/weka/`, accessed on 04/03/2011

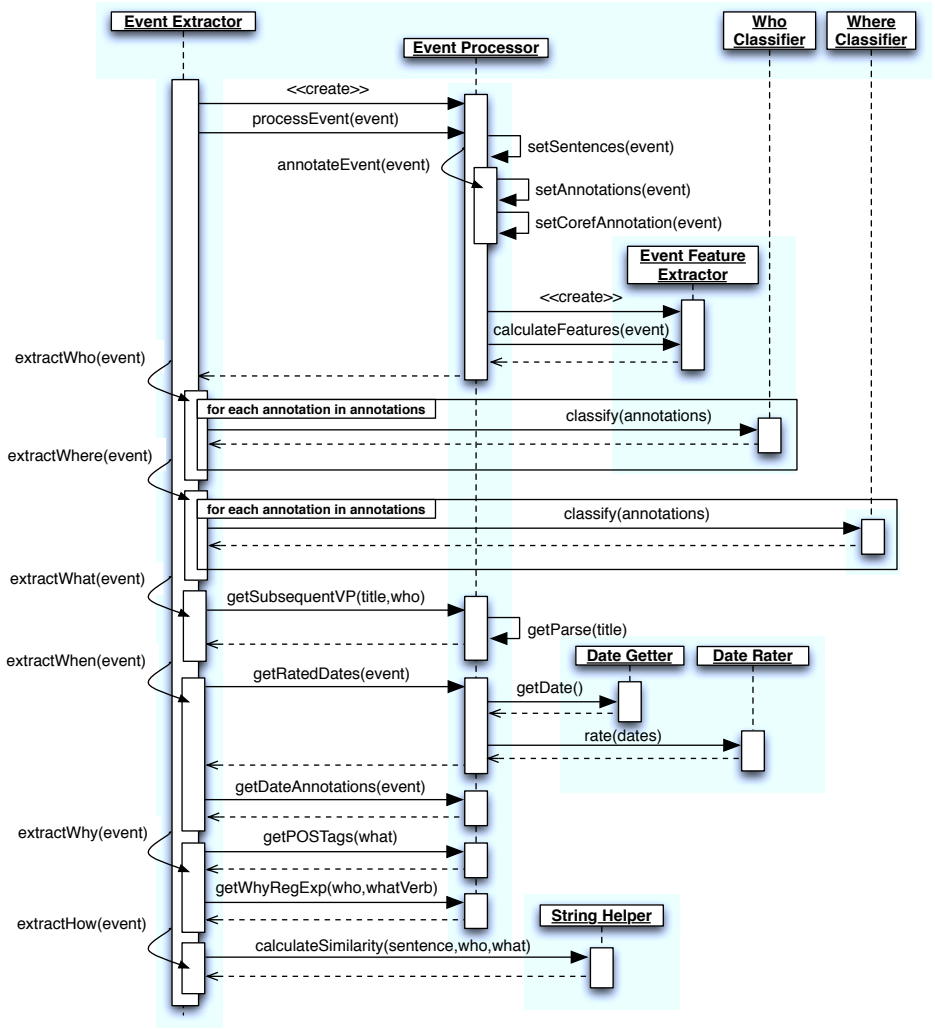



Figure 4.1: Sequence diagram of the 5W1H extraction process

POSTagger, and NER. Furthermore we implemented an OpenNLP -Parser, Phrase Chunker, POSTagger, SentenceDetector and NER. The `NaturalLanguageProcessor` is an abstract class that can be assembled with the different concrete implementations of the NLP tools. In Figure 4.2 one can see the `ws.palladian.preprocessing.nlp` package of Palladian.

Since we focus on a fast and lightweight extraction process, we evaluated execution time and accuracy of POS-Tagger, Phrase Chunker and Sentence Detector. The results can be found in Table 4.1.

We also implemented several Named Entity Recognizer, each capable of annotating Named Entities based on locally stored models or over an online API. A great advantage of online NERs is that they offer a more fine grained distinction between

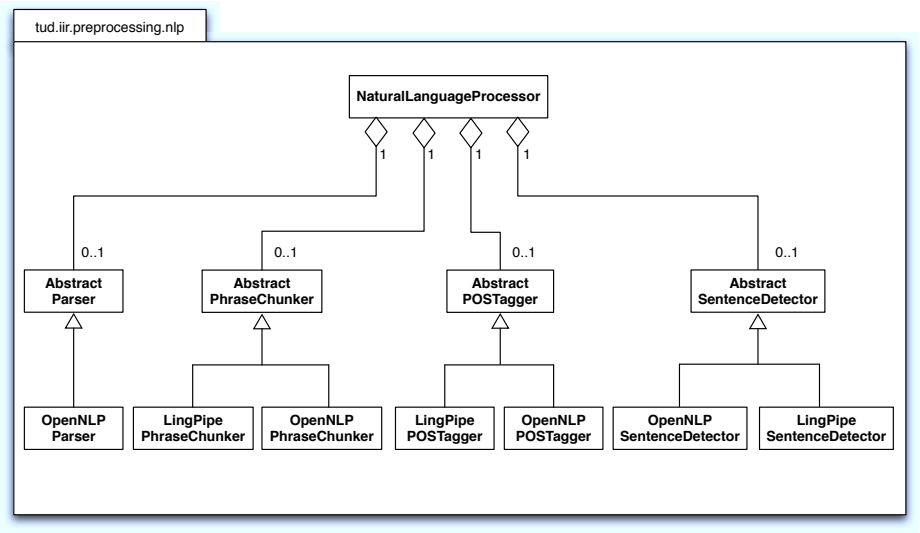



Figure 4.2: Class diagram of the NLP processing package of Palladian

| Implementation | POS-Tagging | Sentence Splitting | Phrase Chunking |
|---|---|---|---|
| OpenNLP | 1 | 1.1 | 1 |
| LingPipe | 1.1 | 1 | 1.1 |

Table 4.1: Execution Time of NLP Tasks in relation

the types of the entities. For instance the Alchemy NER is capable of identifying hundreds of entity types[4]. The OpenCalais NER also offers to annotate a large variety of entity types[5]. Since annotating entities over an online API consumes too much time and requires internet access that we wont to restrict our system to, we decided to focus on local model based NER implementations. We compared three of them that are able to annotate people, locations and organizations and calculated average execution times as listed below in Table 4.2.

[4] `http://www.alchemyapi.com/api/entity/types.html`, accessed on 04/03/2011

[5] `http://www.opencalais.com/documentation/calais-web-service-api/api-metadata/entity-index-and-definitions`, accessed on 04/03/2011

| Implementation | Relative Execution Time |
|---|---|
| OpenNLP 1.4 NER | 1.5 |
| LingPipe NER | 1 |
| Standford NER | 1 |

Table 4.2: Execution Time of Named Entity Recognition in Comparison

The investigation lead us to the following assumptions and decisions.

**POS-Tagging** does not vary in speed much but in accuracy and since we need it for extracting verb phrases from title and important sentences with high accuracy to preserve semantics in `WHAT` and `WHY`, we choose the OpenNLP POS-Tagger.

**Sentence Detection** is for the OpenNLP and LingPipe implementation both fast and accurate, so we just choose the OpenNLP Sentence Detector.

**Parsing / PhraseChunking** is used for extracting our `WHAT` from the topic sentence. As one can see on the passive-active problem 3.3 it is really important to completely extract the semantics belonging to the `WHAT`. This means that not only the verb itself should be recognized with high accuracy but the subsequent phrase as well. Since Phrase Chunking only splits the sentence into phrases, we do not exactly know which phrase belongs to the verb. Most likely it is the subsequent phrase, but we do not know whether it contains all the information to give a complete answer to our question `WHAT`. For this reason we decided to apply parsing that puts out a parse tree which shows the semantic structure of the sentence. With this additional information we can simply refer to the parent node (verb phrase) of the verb and take the whole subtree as our `WHAT`. The difference between parsing and phrase chunking is briefly described in Section 2.2.4.

**Named Entity Recognition** was compared between the OpenNLP, the LingPipe and the Standford NER. The LingPipe and Standford System were nearly equal fast in annotating sample texts. The OpenNLP NER in average needed 150% of the time. We decided to use the LingPipe NER since the alias-i framework also offers Co-reference Resolution which we need to group the belonging entities.

Since we are focusing on a complete and fast extraction of all questions we need to hold the models for the different extraction tasks in storage. Before Extraction starts, the models are loaded into storage which makes the extraction process independent of model size and loading time.

## 4.5 Event Processing

The `EventProcessor` is responsible for the preprocessing of the text and offers functionality for the certain extraction steps of the 5W1H questions. At first we split

the text into sentences and store them within the event object. The next step is to annotate the entities in title and text. In addition to entity annotation we annotate noun phrases (NP) in title with the help of a Phrase Chunker. A Noun Phrase Chunker is able to group POS-tagged noun phrases such as "President/NNP Barack/NNP Obama/NNP" together as one noun phrase "President Barack Obama/NP". Once the entities are annotated we group equal entities by coreference resolution and store the compound of entities within the event object.

For the extraction of `WHO` and `WHERE` we used ML algorithms which require a feature space to classify an extracted entity into relevant or non-relevant with a belonging confidence. The following list describes how the features are being calculated through the `EventFeatureExtractor`.

**The number of occurrences in text** is the size of the co-referenced group of belonging entities.

**The number of occurrences in the headline** is counted with the so called `StringHelper` which already is part of Palladian and simply performs pattern matching with the name of the entity.

**Distribution** is calculated as the average position of the entity relative to the length of the text. The position of each entity is given through a text offset within the entity object.

**Type of the entity** is being detected by the named entity recognizer and is limited to be one of the three: person, organization and location.

For the classification of the `WHERE` candidates we reduce the features space by the type of the entity because we know that the type has to be a location entity.

## 4.6 Event Extraction Process

After the text of the article is preprocessed the gradual extraction of the 5W1H starts. Once the classification models are loaded the `Who-` and `WhereClassifiers` classify the featured entities. Both classifiers return a confidence for each entity held in a list of `rankedCandidates`. The entity with the highest confidence is our most likely `WHO` or `WHERE` and is written into the event object.

The extraction of the `WHAT` is based on the previous extraction of the `WHO`. As one can read in Section 3.3 we are looking for occurrences of `WHO` candidates within the headline via simple pattern matching. If there is no occurrence of a who candidate within the headline, we look for the first occurrence of the highest ranked `WHO` within the text. Once the `WHO` was found, the challenge of this extraction mechanism is to find its subsequent verb phrase. To do this we parse the `WHO` containing sentence with our parser and walk through the parse tree until we find the `WHO`. In detail this means that we look for the node containing our `WHO` and that is a noun. When the node is found we extract the next subsequent node that is a verb or verb phrase.

Here we have a general problem that the subsequent verb phrase in long sentences contains a lot of information that we can not ignore because it is semantically relevant. For the sentence, "A court in Pakistan has sentenced a Muslim prayer leader and his son to life in jail for blasphemy." our subsequent verb phrase is "has sentenced a Muslim prayer leader and his son to life in jail for blasphemy" which indeed is correct but contains a lot more information than we need. In this example sentence even the reason that could fill our `WHY` slot is stated. Finally it is a non-trivial task to filter out the minimal necessary information. We decided to solve this problem by limiting the verb phrase to length. We found out that the length of the verb phrase with enough semantic information to be complete with our `WHO` is 100. A higher number of characters would add unnecessary content to the extracted `WHAT` verb, whereas a too low number might only contain the verb itself. Since our parse tree consists of multiple nodes containing the verb, we select the verb phrase that is shorter than 100 characters.
The following Figures 4.3 and 4.4 of nodes of a parse tree helps to explain the problem of the subsequent verb phrase.

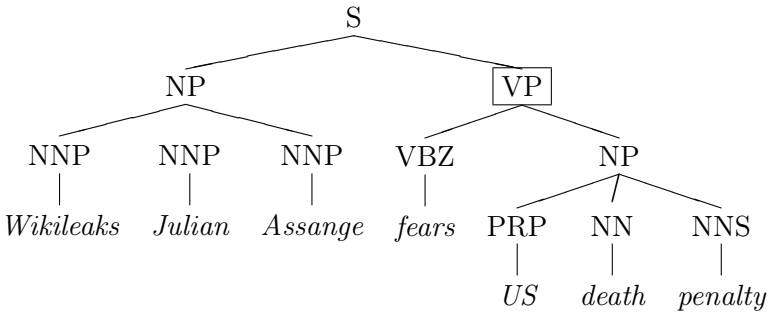



Figure 4.3: Parse tree for an example sentence

For this case the `WHO` is "Julian Assange". As one can see the list holds two candidates for the `WHAT`. On position 6 the verb phrase "fears US death penalty" and on position 7 in the list "fears". The list sorts our verb phrase and its children from long to short. This means the phrases that are a tagged with `V*` are ordered by length. This property enables us to iterate through the list and pick out a long enough verb phrase that contains all necessary information. In the example this is "fears US death penalty". The verb "fears" here does not contain enough information to understand the message.
Our processing chain proceeds with the extraction of `WHEN`. The information about when something happens can be expressed in various ways as it is explained in Section 3.5. We implemented different extraction mechanisms for date and time. Palladian offers date recognition of webpages via various extraction techniques that are based on the properties of the html document. The so called `DateGetter` is used to extract dates from HTML structure, content and the HTML header. The dates can be rated via the `DateEvaluator` that puts out a list that we can treat as `WHEN` candidates that

```
1 Wikileaks' Julian Assange fears US death penalty/S
2 Wikileaks' Julian Assange/NP
3 Wikileaks'/NNP
4 Julian/NNP
5 Assange/NNP
6 fears US death penalty/VP
7 fears/VBZ
8 US death penalty/NP
9 US/PRP
10 death/NN
11 penalty/NN
```

Figure 4.4: Parse tree as list for an example sentence

are ranked through the rating. We take the highest rated date as WHEN.

In addition we implemented an NER capable of annotating dates and time in a given text. For this we use the OpenNLP NER with only models loaded for time and date recognition. The recognized dates are added to the rankedCandidates for the WHEN.

Next is the extraction of WHY which basically is looking for sentences that are indicating a reason for the event itself. To do this we constructed a collection of regular expressions that one can see in Table A.3. Since we also have complex patterns that contain the previously extracted WHAT and the subsequent "to verb" we have to tag the parts of speech of each sentence to identify the verb. If a certain sentence matches a regular expression we store it as a rankedCandidate into the event object. All regular expressions have confidences that we discovered in a previous investigation of 100 news articles as described in Section 3.5. These confidences are stored together with the matching sentences. Finally we take the sentence with the highest confidence to fill the WHY slot.

We found out that the confidence of a sentence to be the reason of the topic is higher if the sentences contains the WHO and WHAT. To extract the HOW from the news article we look for the sentence that contains the WHO and WHAT because this sentence most likely holds additional information about how the event happened. To find this sentence we iterate through all sentences and calculate a similarity between the sentence and the string "WHO WHAT". The equation (3.1) for calculating the similarity is stated in Section 3.7. We rank the sentences by their similarity to the concatenated WHO and WHAT and store them into the event object. The highest ranked sentence fills the HOW slot.

# 5 Results and Evaluation

The following chapter is split into evaluation of the classification algorithms and the evaluation of the NewsX system by a user study. In the first part we explain how we evaluated four different classification algorithms and what are the results. The second part describes the NewsX Evaluation System and comments on the results. Furthermore, we describe origins of errors and possible ways to improve the system.

## 5.1 Evaluation of the Classifiers

This section describes the evaluation of the WHO and WHERE classifier. We introduce the dataset, the evaluation workflow and the discuss results of the evaluation.

### 5.1.1 Dataset

We could not find any pre-tagged dataset that fulfilled our requirements for the classification of WHO and WHERE entities, which lead us to the decision to manually create a set of random news articles. The Dataset consists of 100 articles from the main online news websites that are the BBC[1], the Thomas Reuters news agency[2], the Cable News Network[3], Al Jazeera [4] and the Guardian[5]. For each entry of the dataset we read the article and noted each entity that we consider to be a correct answer to the question for the subject. For an article about "Mobile phone masts linked to mysterious spikes in births" the correct answers for WHO are "masts", "towers" and "transmitter" since they all occur within the article and are related to the same entity.

We then annotated the entities of each news article, performed CR and calculated the features as previously described in section 3.2. To binary classify the entities, we had to decide whether an entity fills our WHO slot or it does not. To do this we iterate through the entities of each news article and compared its name with the previously manually extracted WHOs. Comparison here means that either the name of the entity contains one of the correct answers or one of the correct answers contains the name of the entity. This would for example mark the entity "Mobile phone masts" as match since the word "masts" is included. This automatism generates a list of 2144 entities with calculated features marked as true or false. We used this list to train our WHO classifier.

[1] http://bbc.co.uk/, accessed on 04/03/2011
[2] http://reuters.com/, accessed on 04/03/2011
[3] http://edition.cnn.com/, accessed on 04/03/2011
[4] http://english.aljazeera.net/, accessed on 04/03/2011
[5] http://www.guardian.co.uk/, accessed on 04/03/2011

For evaluating our WHERE classifier we used the same list of extracted and co-referenced entities as for the WHO classifier minus the entities that are not of the type "location". We do this because we assume that the WHERE slot only can be filled by named entities of type "location". This constraint reduces our list of entities to be classified to a size of 513.

### 5.1.2 KNIME

To evaluate our classifiers we used KNIME[6] (Konstanz Information Miner) which is a user-friendly and comprehensive open-source data integration, processing, analysis, and exploration platform that is based on Eclipse[7] technology. The modular architecture of KNIME enables the user to visually create data flows, selectively execute some or all analysis steps and later investigate the results through interactive views on data and models. KNIME offers a plugin that implements the WEKA classification algorithms which are also integrated in Palladian. This enables us to analyze different classification algorithms and parameters before they are finally being implemented. In the previous section we described how we generated a list of features that can be integrated into the KNIME workflow as CSV File. In Figure 5.1 the workflow of the evaluation is visualized.

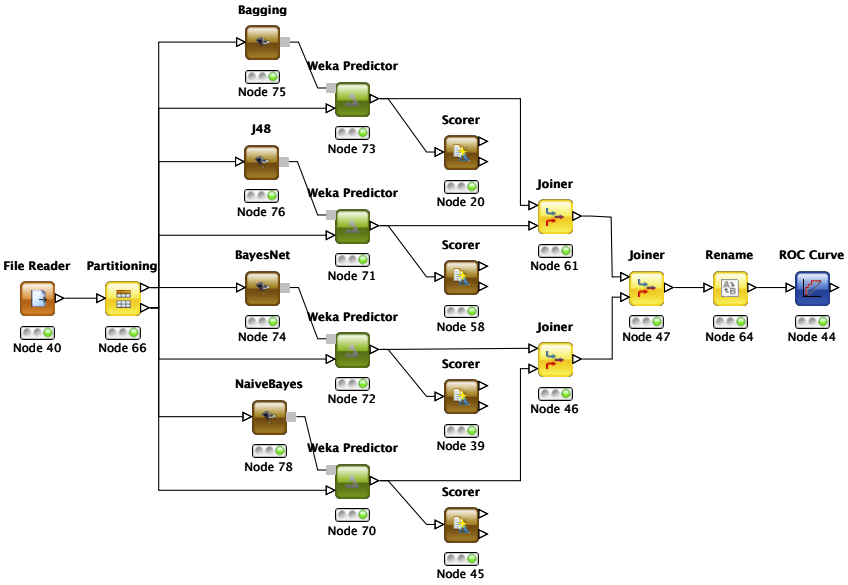



Figure 5.1: KNIME workflow for evaluation of the classifiers

Via a File Reader node the CSV File is read into the workflow. The Partitioning node splits up the data into 2 parts of the same size using a linear sampling strategy.

[6] http://www.knime.org, accessed on 04/03/2011
[7] http://www.eclipse.org, accessed on 04/03/2011

One part is used to train each classifier, the second part is being predicted by the WEKA Predictor based on the trained classifier. The predictor then puts out the classified test data and classification probabilities for the positive value that are used for plotting the ROC Curve. The Scorer node connected to each WEKA Predictor calculates the previously described measures such as precision and recall. The Join nodes combine the results of each predictor and merge them for the ROC curve node.

### 5.1.3 Evaluation Metrics

For evaluating the correctness of a classification approach there are two widely used metrics, *precision* and *recall* that can be seen as extended versions of accuracy. Precision can be seen as a measure of exactness or fidelity whereas recall is a measure of completeness.

In the context of a classification task, the terms *true positives*, *true negatives*, *false positives* and *false negatives* are used to compare the given classification of an item with the desired correct classification. This is illustrated by the Figure 5.2 below:

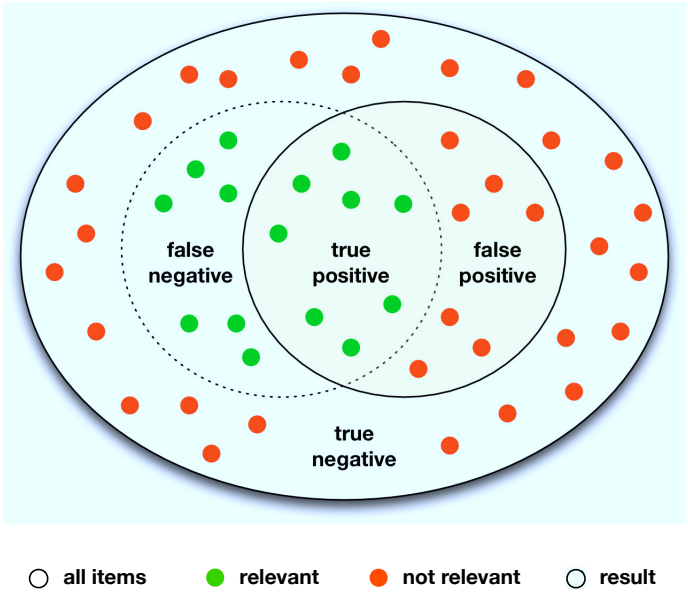



Figure 5.2: Items in classification tasks

A table with two rows and two columns that reports the number of true negatives, false positives, false negatives, and true positives in predictive analytics is called a table of confusion, also known as confusion matrix. Figure 5.3 illustrates this table.

The precision is the number of items correctly labeled as belonging to the positive class divided by the total numbers of elements labeled as belonging to the positive class. Recall in this context is defined as the number of correctly labeled items divided

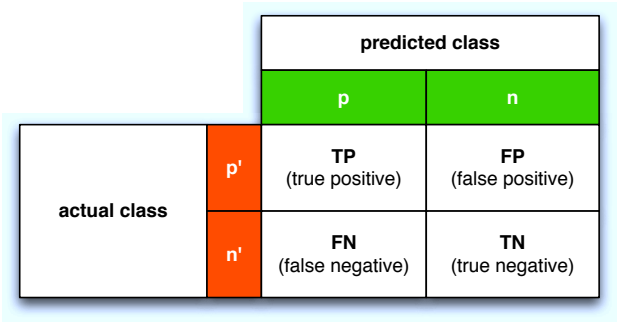



Figure 5.3: Confusion matrix

by the total number of elements that actually belong to the positive class.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{5.1}$$

$$\text{Recall} = \text{True Positive Rate} = \frac{TP}{TP + FN} \tag{5.2}$$

For the sake of completeness we here give the equations for two other related measures, *Specificity*(5.3) and *Accuracy*(5.4).

$$\text{Specificity} = \text{True Negative Rate} = \frac{TN}{TN + FP} \tag{5.3}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.4}$$

The so called *F-Measure* or *F-Score* is the harmonic mean of precision and recall and combines the two measures. The calculation is shown in Equation (5.5).

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}} \tag{5.5}$$

With $\beta$ precision and recall can be weighted. For example in $F_1$ measure precision and recall are evenly weighted as one can see in Equation (5.6).

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{5.6}$$

Two other commonly used $F$ measures are the $F_2$ measure, which weights recall twice as much as precision, and the $F_{0.5}$ measure, which weights precision twice as much as recall.

For the classification of our entities a relevant entity is marked as 1 whereas a not relevant entity is marked as 0. A true positive result of our classification is when an entity that is our who is correct classified with 1.

In classification evaluation the *Receiver operating characteristic* (ROC) *curve* is a fundamental tool. The ROC curve describes the performance of a model across the entire range of classification thresholds. All ROC curves begin in the bottom-left corner and rise to the top-right corner. Each point on the curve is created by plotting the unique *true positive rate* (TPR) and *false positive rate* (FPR) associated with each unique test value.

The area under the ROC curve (AUC) is a measure of performance that can be used to compare classification results. The larger the AUC, the more accurate the classification.

### 5.1.4 Results

We compared four classification approaches, a Bayes Network, a Decision Tree (J48), a Bootstrapping (Efron and Tibshirani, 1993) Bagging (Breiman, 1996) and a Naïve Bayes 2.1.2. For classification we split our 2144 entities into randomly chosen 50% training set and 50% test set. On our training set of 2144 `WHO` Candidates the Bagging Classifier performs best. In Table A.4 one can find the results of the different classifiers compared. The following Figure 5.4 plots the ROC curves of the different classifiers. As one can see the Bagging Classifier has the highest AUC.

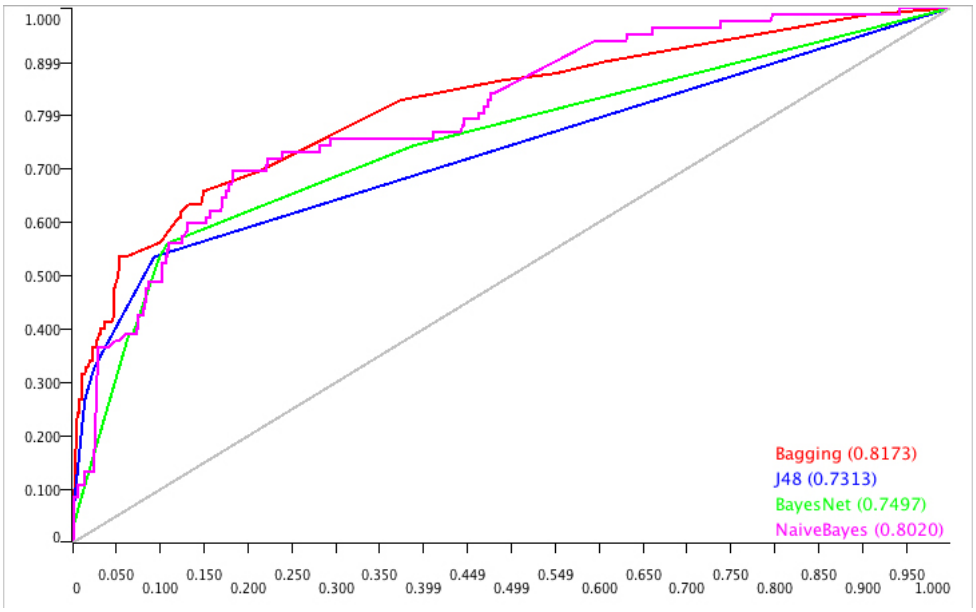



Figure 5.4: ROC curves of Who classifiers

In Figure 5.4 there are two curves that only have one or two changes of their slope. This can be traced back to the graph based character of the network (Bayes Net) and tree (J48), which are discrete classifiers converted to scoring classifiers. Each change of the slope represents a node in the tree or network. Since we are using a small feature space the probabilistic classifiers perform better than the converted discrete

classifiers.
At a certain point the curves obviously change their slope showing a drift off to the right hand side which implies a high number of false positives caused by the high number of negative instances.
For the classification of the entities there is room for improvement. A first step for better classification results is to increase the training data, because more training data means a more precise prediction model. Another point is to analyze the groups of entities and make more restrictions on the entities being selected, which means to decrease the entities to be classified for each document.
Taking the probabilities of the Named Entity Recognizer into consideration is also worth to be investigated. Instead of simply taking the type of the entity as a feature we could calculate a combined feature derived from the probabilities of the recognition and the resulting type.
The selection of the correct entity by classification can only be seen as a starting point on the way to a precise answer to the questions `WHO` and `WHERE`. Constructing a final answer that fits the remaining questions requires further semantic processing which we propose in section 5.2.1.

## 5.2 User Study of the NewsX system

In this section, we evaluate our event extraction mechanism and the quality of the extracted 5W1H tuples. Extraction of semantic information from human language is difficult to evaluate, since it often needs human interpretation to decide if the extracted data fits the defined requirements. In our case of extracting the answers to the 5W1H questions there is no certain solution for each answer but a broad range of possible solutions that require user supported evaluation. Especially for answering the questions `WHY` and `HOW` it is necessary to understand the news article as a whole and to manually decide whether the answer to the question is mentioned within the text. These are the reasons that made it necessary to evaluate our system NewsX through a user study.

To evaluate our system we developed a web-based evaluation tool, that enabled a user to read a news article in an neutral environment and to decide whether the extracted 5W1H are right, wrong or partially correct. Figure 5.5 shows an example from the evaluation website.

The evaluation website was implemented via the PHP-based CakePHP[8] Framework. It allows the rapid development of web applications using the concept of Model-View-Controller.

Before evaluation we extracted current news articles from the same news websites that we used for constructing our dataset for classification. We sorted the news article randomly and extracted the 5W1H from all of them. The users are introduced to the

[8] `http://cakephp.org/`, accessed on 04/03/2011

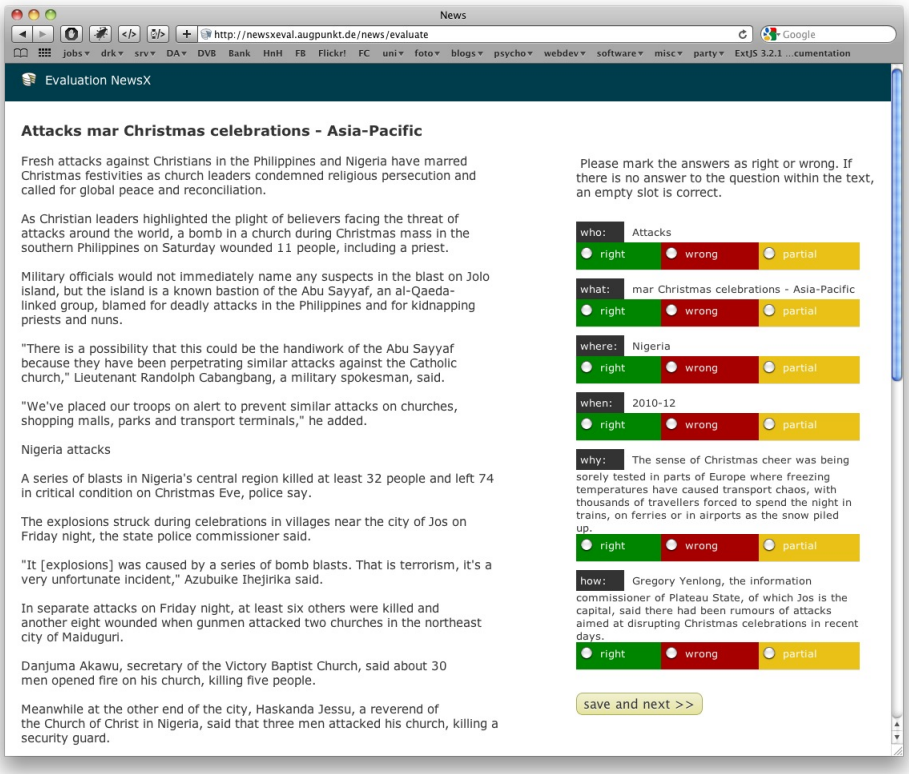



Figure 5.5: Screenshot of the evaluation system

evaluation and directed to mind the semantic relation between the answers, especially between the subject and verb. Furthermore we gave the advice that an empty answer can be correct, if there is no answer to the question within the text.
Each user that logged in with an e-mail address evaluated the same news article and extracted 5W1H. The user was expected to read through the article and then to decide if the extracted answers to the 5W1H are wrong, right or partially correct. We introduced to the users that they should select partial if the slot in their opinion only contains a part of the correct answer or they are uncertain. This enables us to distinguish between explicit right or wrong extracted answers and answers that need further analysis to improve the system.

### 5.2.1 Results

Within one week 12 different users evaluated 56 news articles. We have to mention, that we removed extracted news articles with less then 5 sentences since we assume that they do not contain enough information to answer the 5W1H. This limitation reduced the number of extractions by two. To achieve a high reliability we let multiple users evaluate the same news articles. In Figure 5.6 the number of user evaluations

per news article is illustrated. As one can see, there are 23 articles that were evaluated from more than one user, whereas the remaining 23 articles were evaluated by only one person. To decide how reliable the evaluations from only one user are, we calculated the *Reliability of Agreement* for the the evaluations with more than two users as described in the following section.

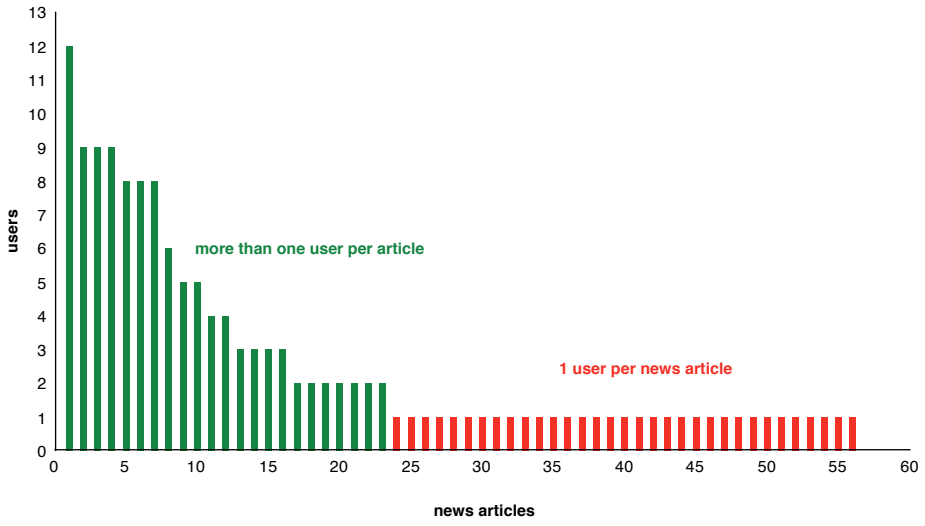



Figure 5.6: Number of user evaluations per news article

### Reliability of Agreement

Since we have one or more user, that evaluated the extraction for one news article, we are interested in the the reliability of agreement between the so called "raters". *Fleiss' Kappa* is a statistical measure for assessing the reliability of agreement or inter-rater reliability between a number of raters when assigning categorical ratings to a number of items. The measure calculates the degree of agreement in classification over that which would be expected by chance and is scored as a number between 0 and 1. The measure can be interpreted as the extend to which the observed amount of agreement among raters exceeds what would be expected if all raters made their ratings completely randomly. The kappa, $\kappa$, can be defined as,

$$\kappa = \frac{\bar{P} - \bar{P}_e}{1 - \bar{P}_e} \tag{5.7}$$

.

The factor $\bar{P} - \bar{P}_e$ gives the degree of agreement actually achieved above chance, whereas the factor $1 - \bar{P}_e$ measures the degree of agreement attainable over and above what would be predicted by chance. Complete agreement of the raters results in $\kappa = 1$. If there is no agreement among the raters then $\kappa \leq 0$.

Using the Fleiss' Kappa statistics requires that each news article is being evaluated by the same number of raters. Since we have different amount of raters for each article, we have to modify the formula. The different amount of raters per questions only affects the calculation of the probability of random agreement $P_e$ which is the agreement by chance. To shortcut the calculation we assume that the proportion of all assignments to the categories is uniformly distributed, thus the mean proportion of agreement is $\frac{1}{3}$.

Another way to make our data fitting the requirements of Fleiss' Kappa is to normalize the data to a number of 6 raters. Applying the formula to the normalized data results in a mean proportion of agreement that is slightly worse than the the agreement assuming uniform distribution.

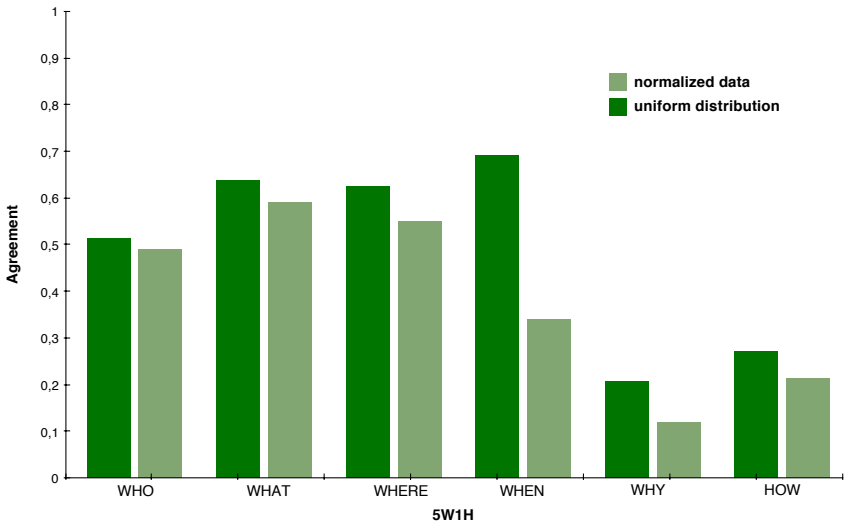



Figure 5.7: Agreement among users

In Figure 5.7 the compared agreement is illustrated. As one can see, the agreement is nearly equal for normalized data and assumption of uniform distribution of random agreement, except the agreement among users on `WHEN` extraction. The reason for this difference is the high number of "right" ratings that changes the probability of random agreement and this affects the $\kappa$ value. Landis and Koch (1977) gave the Table 5.1 for interpreting $\kappa$ values. In general the agreement is not reliable enough and strongly differs between the 5W1H to trust the evaluations only made by one rater. Despite this we do not want to reject the results, so we make the distinction between all ratings and ratings with more than one user. In Figure 5.8 the results of the evaluation under this prospect are illustrated.

| $\kappa$ | Interpretation |
|---|---|
| $< 0$ | Poor agreement |
| $0.0 - 0.2$ | Slight agreement |
| $0.21 - 0.40$ | Fair agreement |
| $0.41 - 0.60$ | Moderate agreement |
| $0.61 - 0.80$ | Substantial agreement |
| $0.81 - 1.00$ | Almost perfect agreement |

Table 5.1: Interpretation of $\kappa$ values

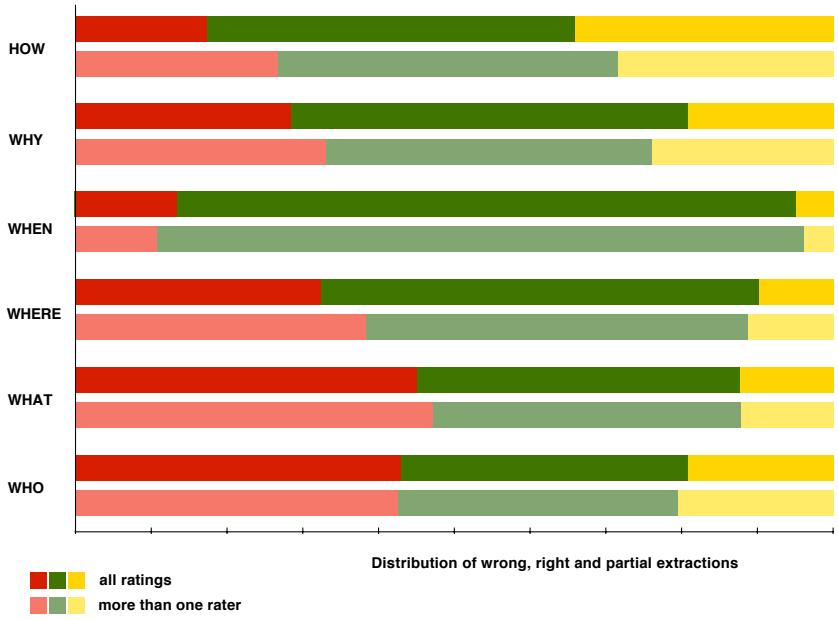



Figure 5.8: Results of the 5W1H user study

### Confidence

To describe how reliable our results are, we calculated the confidence interval of our measurements. A confidence interval is an interval in which a measurement of trial falls corresponding to a given probability. To calculate the confidence intervals we need to calculate the standard error (S.E.), which is a method of estimating the standard deviation of our sample distribution. The general expression for approximate $N\%$ confidence levels for an error is shown in equation (5.8), where $\theta$ is the sample error and $n$ is the sample size.

$$S.E. = \sigma(p) = z_N \sqrt{\frac{\theta * (1 - \theta)}{n}} \tag{5.8}$$

For our `WHO` extraction we have a sample set that contains 56 news articles. We

define an extraction as wrong, if more than the half of raters marked it as wrong, which results in 23 wrong extractions. The sample error for the WHO extractions is $\theta = 23/56 = 0.41$. The constant $z_N$ is chosen depending on the desired confidence level. We chose a confidence level of 90% where the constant $z_N$ is 1.64. The resulting confidence level for our WHO extraction is $S.E. = 0.41 \pm 0.11$.
We calculated the confidence intervals for each question of the 5W1H and investigated two sample sets. One set includes only the news articles that were evaluated from more than two users and the other set consists of all news articles. The error rates and belonging confidences are shown in Figure 5.9.

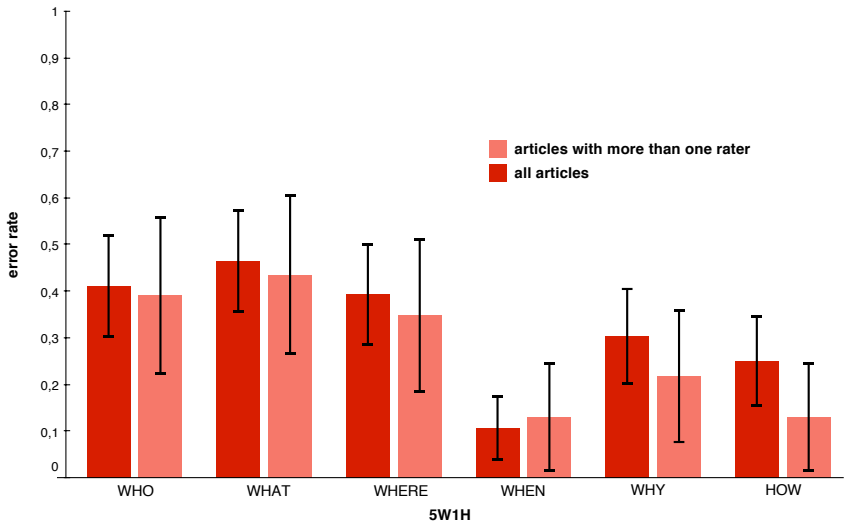



Figure 5.9: Error rates with confidence intervals

The black line segments are illustrating the confidence intervals. The confidence intervals of the error rates covering all raters are slightly smaller than the intervals of the sample set with news articles evaluated by more than one rater. The reason is that larger underlying sample set results in higher confidence, thus shorter bars.

### 5.2.2 Discussion

The following discussion describes the relationships and generalizations that we found out. Furthermore, it explains the causes of problems and gives recommendations to improve the results.

An observation that can be made is that the results of WHO and WHAT extraction are nearly equal. The reason is that the extraction process of WHO and WHAT are closely coupled. We assume that improvement of the WHO extraction will lead to better results of WHAT extraction. The high percentage of partial rates has its origin in the coarse grained extraction. There are multiple cases, where the extracted WHO is partially correct because it is incomplete or contains verbs that were wrongly identified as

nouns, which has an high impact on the result. For example the phrase "Queen praises sport" was identified as noun phrase, because the POS-tagger tagged "praises" as noun instead of verb. On the other hand the identification of the noun phrase worked well for "Kabul raid" and subsequent verb phrase "kills two after U.S. embassy threat". A promising approach to improve the extraction concerning this issue is to analyze the sequence of parts of speech to derive new rules for clearly separating the subject and verb of the topic sentences. Here the confidences of the tagged words can be helpful. To reduce the negative instance for the `WHO` classification, one might try to apply NER on the first sentences and title, since according to the character of news articles the subject is mentioned within this part of the article. Furthermore, this would reduce the time needed for recognizing the entities.

The results for the extraction of `WHERE` are satisfying, since half of the extractions are correct. The larger number of wrong extracted locations can be traced back to wrong assignment of entity type "location" through the NER, that leads to extracted entities that are no locations but marked as those. The other origin of wrong and partial extractions is the coarse grained localization of the event. Most news articles contain information about the country or continent where the event happened, which most likely is selected by the classifier to fill the `WHERE` slot, but not necessarily is the best possible answer to the question. When within the article a more specific location is mentioned, the users marked the extraction of the coarse localization as wrong or at least partial. In general the process of extracting the location can be improved by enlarging the training set and adding the confidence of the recognized entity as feature for the classification. Another feature for classification could be the granularity of the recognized location, but this would require further processing and time.

Answering the question `WHEN` works best with about 80% correct extracted time specifications. The good result is caused by the structure of the document, that in the most cases contains a concrete time at the begin of the document or the header, which easily could be extracted through the `DateGetter`. The wrong or partial marked extractions have their reason in date or relative time mentions such as "yesterday" or "last monday" within the text. Here the problem is to ensure the relation between the time mention and the event. To resolve this it requires higher semantic interpretation of the sentences including the time references, that is called recognition of temporal expressions and time periods.

The extraction of `WHY` and `HOW` was satisfying. The equal distribution of wrong, right and partial can be traced back to the difficult to understand semantic relations between the document and the answer to the questions. This means that the user needs full understanding of the whole text to decide wether the selected sentence actually answers the question `HOW` or `WHY`. Source of errors are the wrong relation between the extracted subject of the article and the extracted `WHY` as reason for the subject. The regular expression to find signal words indicating such a reason

are not necessarily selecting the sentence that is reason of the subject. So a higher involvement of the previously extracted `WHO` and `WHAT` could lead to better results. To improve the set of regular expressions that are aimed to find the cause indicating sentence, it can be conductive to apply a machine learning approach on these patterns. Treating a topic sentence as `HOW` seems to be a promising approach to answer the question, but there is still room for improvement especially concerning the relation to the extracted subject.

On one hand we can say it is highly relevant to ensure the semantic relation between the given answers by involving the previously extracted data. On the other hand this means, an error within the extraction chain possibly affects the whole extraction of the 5W1H. To resolve this, the idea of collecting multiple news articles for one topic is a promising approach.

# 6 Conclusion and Future Work

This chapter concludes the thesis with a summary of the contributions to our research and proposes several topics that should be considered by future work.

## 6.1 Conclusion

This thesis was aimed to develop an event extraction system capable of extracting the answers to the 5W1H questions from a news article. We wanted to investigate the character of needed information within news articles and to derive regularities that can be conductive to extract them. A further goal was to find appropriate mechanisms from the research field of Natural Language Processing and Information Extraction to find and extract the 5W1H. We introduced the tasks, applications and approaches of Machine Learning with main focus on classification problems. In Chapter 2.2 we examined important techniques from the field of Natural Language Processing that are valuable for extracting parts of free text. We studied various approaches of event extraction that interact with different fields of research such as Question Answering and Text Summarization. From pattern-based approaches over event-based summarization of news clusters to recently applied Semantic Role Labeling of topic sentences we could not find an approach that fulfilled our requirements of a domain independent, lightweight and document-based event extraction system. The two systems that were supportive and close to our requirements are introduced in Chapter 2.4. In the conceptual stage we developed a strategy of gradual detection of the 5W1H that we prototypically implemented as the NewsX system.

The main contributions are summarized in the following paragraphs:

- We revealed properties of the needed information based on the character of news articles that are valuable for extracting the 5W1H as event description.
- We investigated multiple mechanisms from related fields of research and selected the appropriate techniques to extract an event description from a domain independent news article.
- We developed an extraction system of gradual detection of the 5W1H, that uses classification of previously recognized entities to extract the `WHO` and `WHERE`. Based on the determined features of entities, we compared different classification algorithms and have chosen a Bagging Classifier to select the `WHO` and `WHERE` from the recognized entities.

- To find the appropriate WHAT we extracted the subsequent verb phrase of the previously extracted WHO. The sentence based extraction of phrases required to deal with semantics of the sentence. Due to error-prone POS-tagging it was not trivial to find the matching verb phrase that is elaborate enough to understand the topic of the event.
- The high degree of intelligence that is needed to extract the answer to the question WHY and HOW something happens, forced us to apply a coarse grained extraction method for these answers. However, the approach of looking for words that indicate causal relations lead to satisfying results for WHY extraction. To answer the question HOW we looked for the sentence that describes the event briefly, which as well lead to promising results.
- We also deliver a list of ranked candidates for each answer that we used to optionally change an answer to keep the semantic relation. The list can also be assistant to further processing.
- In contrast to sentence based event extraction systems that basically bear on information within one sentence to extract event facts, we have to solve the problem of the scattered information. This means we had to ensure the semantic relation between the extracted parts. To give semantically related answers to the 5W1H, it requires semantic interpretation of parts of the news article, especially for the strongly coupled WHO and WHAT that we treat as subject and verb.
- We find that the most important factor that affects the correctness of 5W1H is the complexity of language, which makes the extractions on sentence level error-prone.
- The gradual extraction is highly significant, since it improves the results among the extracted answers.

In general a more precise extraction needs deeper semantic processing and solution of the fine grained problems, thus more processing time and logic. It depends on the use case whether a deep and accurate extraction is wanted or a fast coarse grained extraction is enough.

## 6.2 Future Work

Besides the improvements suggested in the previous chapter, during the thesis some ideas for possible extensions and improvements evolved. There are several lines of research arising from this work which should be pursued.

- The thesis was aimed to develop a lightweight extraction system. It became obvious that an orthographically and semantically correct extraction requires deeper semantic processing such as semantic parsing and recognition of temporal

expressions. With a customizable extraction chain of different extraction mechanisms the depth of extraction would become scalable, which allows to adapt the extraction process to specific requirements. Furthermore, the implementation of different extraction techniques allows to research the impact on event extraction quality.

- To improve the reliability of the extracted event facts, the aggregation of further news articles describing the same event should be taken into consideration. We implemented a prototypical `EventAggregator` that aggregates event URLs over a web search engine by a constructed query string. The challenge here is to only aggregate appropriate news articles and to combine the extracted 5W1Hs into one with high confidence.

- The machine learning approach for the classification of recognized entities approved well. Constructing a larger dataset and annotating the event facts opens the door to further applications of Machine Learning as for the causal relation indicating regular expressions and temporal references.

In summary, the thesis provided a brief overview of the current trends and challenges in Event Extraction and proofed the suitability of the introduced approaches. With the previously described points we made proposals for further improvements of the NewsX Event Extraction System.

# A Tables

## A.1 Extract from the tag-set of the Brown Corpus

| Tag | Description | Examples |
|---|---|---|
| . | sentence closer | . ; ? ! |
| ( | left paren | |
| ) | right paren | |
| * | not, n't | |
| – | dash | |
| , | comma | |
| : | colon | |
| ABL | pre-qualifier | quite, rather |
| ABN | pre-quantifier | half, all |
| ABX | pre-quantifier | both |
| AP | post-determiner | many, several, next |
| AT | article | a, the, no |
| NN | singular or mass noun | |
| NN$ | possessive singular noun | |
| NNS | plural noun | |
| NNS$ | possessive plural noun | |
| NP | proper noun or part of name phrase | |
| NP$ | possessive proper noun | |
| NPS | plural proper noun | |
| NPS$ | possessive plural proper noun | |
| RB | adverb | |
| RBR | comparative adverb | |
| RBT | superlative adverb | |
| RN | nominal adverb | here then, indoors |
| RP | adverb/particle | about, off, up |
| und TO | infinitive marker to | |
| UH | interjection, exclamation | |
| VB | verb, base form | |
| VBD | verb, past tense | |
| VBG | verb, present participle/gerund | |
| VBN | verb, past participle | |
| VBZ | verb, 3rd. singular present | |

## A.2 ACE07 Event Types and Subtypes

| Types | Subtype |
|---|---|
| Life | Be-Born, Marry, Divorce, Injure, Die |
| Movement | Transport |
| Transaction | Transfer-Ownership, Transfer-Money |
| Business | Start-Org, Merge-Org, Declare-Bankruptcy, End-Org |
| Conflict | Attack, Demonstrate |
| Contact | Meet, Phone-Write |
| Personnel | Start-Position, End-Position, Nominate, Elect |
| Justice | Arrest-Jail, Release-Parole, Trial-Hearing, Charge-Indict, Sue, Convict, Sentence, Fine, Execute, Extradite, Acquit, Appeal, Pardon |

## A.3 Regular Expressions for the extraction of Why

| Regular Expression | Confidence |
|---|---|
| (WHAT(.*)to/TO (.*)/VB) | 0.5 |
| (WHAT(.*)will) | 0.5 |
| (since) | 0.2 |
| (cause) | 0.3 |
| (because) | 0.3 |
| (hence) | 0.2 |
| (therefore) | 0.3 |
| (why) | 0.3 |
| (result) | 0.4 |
| (reason) | 0.3 |
| (provide) | 0.1 |
| (s behind) | 0.2 |
| (Due to) | 0.2 |

## A.4 Results for the Who Classification

| Classifier | Precision | Recall | F-Measure | AUC |
|---|---|---|---|---|
| Bagging | 80.6% | 33.1% | 47% | 0.8173 |
| Bayes Network | 33.7% | 39.3% | 32.3% | 0.7497 |
| Neural Network (J48) | 78.2% | 26.4% | 36.3% | 0.7313 |
| Naïve Bayes | 51.6% | 18.4% | 27.1% | 0.8020 |

# List of Figures

# List of Tables

# List of Abbreviations

**5W1H** Who, What, Where, When, Why and How

**ACE** Automated Content Extraction

**AI** Artificial Intelligence

**API** Application Programming Interface

**AUC** Area Under Curve

**CFG** Context-free Grammars

**CNFE** Chinese News Fact Extractor

**CSV** Comma-separated values

**EE** Event Extraction

**HMM** Hidden Markov Models

**HTML** Hypertext Markup Language

**IE** Information Extraction

**IR** Information Retrieval

**ME** Maximum Entropy

**ML** Machine Learning

**MUC** Message Understanding Conference

**NER** Named Entity Recognition

**NEXUS** News cluster Event eXtraction Using language Structures

**NLP** Natural Language Processing

**NP** Noun Phrase

**CR** Co-reference Resolution

**PCFG** Probabilistic Context Free Grammars

**POS** Parts of Speech

**ROC** Receiver operating characteristic

**SBD** Sentence Boundary Disambiguation

**SRL** Semantic Role Labeling

**SVM** Support Vector Machines

**URL** Uniform Resource Locator

**VDR** eVent Detection and Recognition

**VP** Verb Phrase

# Bibliography


Eugene Agichtein, Luis Gravano, Jeff Pavel, Viktoriya Sokolova, and Aleksandr Voskoboynik. Snowball: a prototype system for extracting relations from large text collections. *SIGMOD Rec.*, 30(2):612, 2001. ISSN 0163-5808. 19

David Ahn. The stages of event extraction. In *ARTE '06: Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, pages 1–8, Morristown, NJ, USA, 2006. Association for Computational Linguistics. ISBN 1-932432-81-7. 20

Bengt Altenberg. Causal linking in spoken and written english. 1984. 34

Martin Atkinson, Jakub Piskorski, Bruno Pouliquen, Ralf Steinberger, Hristo Tanev, and Vanni Zavarella. Online-monitoring of security-related events. In *COLING '08: 22nd International Conference on on Computational Linguistics: Demonstration Papers*, pages 145–148, Morristown, NJ, USA, 2008. Association for Computational Linguistics. 19

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. The berkeley framenet project. In *Proceedings of the 17th International Conference on Computational Linguistics*, pages 86–90, Montreal, Quebec, Canada, 1998. Association for Computational Linguistics. 20

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. Open information extraction from the web. In *IJCAI'07: Proceedings of the 20th international joint conference on Artifical intelligence*, pages 2670–2676, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc. 19

Taylor Booth and Richard Thompson. Applying probability measures to abstract languages. *IEEE Trans. Comput.*, 22:442–450, May 1973. ISSN 0018-9340. 17

Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996. 31, 49

Francesca Carmagnola. The five ws in user model interoperability. In *Workshop on Ubiquitous User Modeling (UbiqUM), Intelligent User Interfaces (IUI) Conference*, 2008. 6

Zheng Chen and Heng Ji. Graph-based event coreference resolution. In *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing (TextGraphs-4)*, pages 54–57, Suntec, Singapore, August 2009. Association for Computational Linguistics. 20

Thomas G. Dietterich. Machine-learning research – four current directions. *AI MAGAZINE*, 18:97–136, 1997. 11

Bonnie Dorr, David Zajic, and Richard Schwartz. Hedge trimmer: a parse-and-trim approach to headline generation. In *Proceedings of the HLT-NAACL 03 on Text summarization workshop - Volume 5*, HLT-NAACL-DUC '03, pages 1–8, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. 30

Bradley Efron and Robert Tibshirani. An introduction to the bootstrap, 1993. 31, 49

Oren Etzioni, Michael Cafarella, Doug Downey, Stanley Kok, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. Web-scale information extraction in knowitall: (preliminary results). In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 100–110, New York, NY, USA, 2004. ACM. ISBN 1-58113-844-X. 19

Elena Filatova and Vasileios Hatzivassiloglou. Event-based extractive summarization. In Stan Szpakowicz Marie-Francine Moens, editor, *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 104–111, Barcelona, Spain, July 2004. Association for Computational Linguistics. 22

Charles Fillmore, Srini Narayanan, and Collin Baker. What can linguistics contribute to event extraction? In *Proceedings of the workshop on Event Extraction and Synthesis, held in conjunction with the AAAI 2006 conference*. American Association for Artificial Intelligence, 2006. 24

Sanda Harabagiu Finley and Sanda M. Harabagiu. Generating single and multi-document summaries with gistexter. In *In U. Hahn D. Harman (Eds.), Proceedings of the workshop on automatic summarization*, pages 30–38, 2002. 23

W. Nelson Francis and Henry Kucera. Brown corpus manual. Technical report, Department of Linguistics, Brown University, Providence, Rhode Island, US, 1979. URL `http://icame.uib.no/brown/bcm.html`. 16

Dayne Freitag and Andrew K. Mccallum. Information extraction with hmms and shrinkage. In *In Proceedings of the AAAI-99 Workshop on Machine Learning for Information Extraction*, pages 31–36, 1999. 12

Roger Garside, Geoffrey Leech, and Anthony McEnery. *Corpus Annotation*. Longman, London and New York, 1997. 16

Martin Gregor. Altersbestimmung von webseiten. Master's thesis, Dresden University of Technology, 2010. 33

Ralph Grishman, Silja Huttunen, and Roman Yangarber. Real-time event extraction for infectious disease outbreaks. In *Proceedings of the second international conference on Human Language Technology Research*, pages 366–369, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc. 19

Darrell C. Ince, editor. *Mechanical intelligence (collected works of A. M. Turing)*. North-Holland Publishing Co., Amsterdam, The Netherlands, The Netherlands, 1992. ISBN 0-444-88058-5. 13

Takashi Inui, Kentaro Inui, and Yuji Matsumoto. Acquiring causal knowledge from text using the connective marker tame. 4:435–474, December 2005. 34

Final Report James, James Allan, Jaime Carbonell, George Doddington, Jonathan Yamron, Yiming Yang, James Allan Umass, Brian Archibald Cmu, Doug Beeferman Cmu, Adam Berger Cmu, Ralf Brown Cmu, Ira Carp Dragon, George Doddington Darpa, Alex Hauptmann Cmu, John Lafferty Cmu, Victor Lavrenko Umass, Ron Papka Umass, Jay Ponte Umass, and Mike Scudder Umass. Topic detection and tracking pilot study. In *In Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, pages 194–218, 1998. 19

Heng Ji. Cross-lingual predicate cluster acquisition to improve bilingual event extraction by inductive learning. In *UMSLLS '09: Proceedings of the Workshop on Unsupervised and Minimally Supervised Learning of Lexical Semantics*, pages 27–35, Morristown, NJ, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-34-3. 20

Heng Ji and Ralph Grishman. Refining event extraction through cross-document inference. In *Proceedings of ACL-08: HLT*, pages 254–262, Columbus, Ohio, June 2008a. Association for Computational Linguistics. 20

Heng Ji and Ralph Grishman. Refining event extraction through cross-document inference. In *Proceedings of ACL-08: HLT*, pages 254–262, Columbus, Ohio, June 2008b. Association for Computational Linguistics. 21

Heng Ji, Ralph Grishman, Zheng Chen, and Prashant Gupta. Cross-document event extraction and tracking. In *Recent Advances in Natural Language Processing*, 2009. 20

Karen Spärck Jones. Automatic summarising: The state of the art. *Information Processing  Management*, 43(6):1449–1481, 2007. 22

Gary King and Will Lowe. An automated information extraction tool for international conflict data with performance as good as human coders: A rare events evaluation design, international organization. In *International Organization, Vol. 57, No. 3*, pages 617–642, 2003. 19

Paul Kingsbury and Martha Palmer. Propbank: the next level of treebank. In *Proceedings of Treebanks and Lexical Theories*, 2004. 20

Miroslav Kubat. Neural networks: a comprehensive foundation by simon haykin, macmillan, 1994, isbn 0-02-352781-7. *Knowl. Eng. Rev.*, 13(4):409–412, 1999. ISSN 0269-8889. 13

Richard Landis and Gary Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33:159–74, March 1977. 53

Raymond Lau, Roni Rosenfeld, and Salim Roukos. Adaptive language modelling using the maximum entropy approach. In *Proceedings ARPA Human Language Technologies Workshop*, March 1993. 11

Wenjie Li, Mingli Wu, Qin Lu, Wei Xu, and Chunfa Yuan. Extractive summarization using inter- and intra-event relevance. In *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 369–376, Morristown, NJ, USA, 2006. Association for Computational Linguistics. 22, 34

Shasha Liao and Ralph Grishman. Using document level cross-event inference to improve event extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 789–797, Morristown, NJ, USA, 2010. Association for Computational Linguistics. 22

Maofu Liu, Wenjie Li, Mingli Wu, and Qin Lu. Extractive summarization based on event term clustering. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 185–188, Prague, Czech Republic, June 2007. Association for Computational Linguistics. 22, 23, XI

Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, 1999. ISBN 0262133601. 22

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008. ISBN 0521865719, 9780521865715. 10, 14

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: the penn treebank. *Computational Linguistics*, 19(2):313–330, 1993. ISSN 0891-2017. 16

Elaine Marsh and Dennis Perzanowski. Muc-7 evaluation of ie technology: Overview of results. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*, 1998. 20

Nancy McCracken. Combining techniques for event extraction in summary reports. In *Proceedings of the workshop on Event Extraction and Synthesis, held in conjunction with the AAAI 2006 conference*. American Association for Artificial Intelligence, 2006. 24

Kathleen McKeown, Regina Barzilay, John Chen, David K. Elson, David K. Evans, Judith Klavans, Ani Nenkova, Barry Schiffman, and Sergey Sigelman. Columbia's newsblaster: New features and future directions. In *HLT-NAACL*, 2003. 23

Harvey J. Murff, Vimla L. Patel, George Hripcsak, and David W. Bates. Detecting adverse events for patient safety research: a review of current methodologies. *J. of Biomedical Informatics*, 36(1/2):131–143, 2003. ISSN 1532-0464. 19

Martina Naughton, Nicola Stokes, and Joe Carthy. Investigating statistical techniques for sentence-level event classification. In *COLING '08: Proceedings of the 22nd International Conference on Computational Linguistics*, pages 617–624, Morristown, NJ, USA, 2008. Association for Computational Linguistics. ISBN 978-1-905593-44-6. 20, 22

Tomoko Ohta, Yoshimasa Tsuruoka, Jumpei Takeuchi, Jin-Dong Kim, Yusuke Miyao, Akane Yakushiji, Kazuhiro Yoshida, Yuka Tateisi, Takashi Ninomiya, Katsuya Masuda, Tadayoshi Hara, and Jun'ichi Tsujii. An intelligent search engine and gui-based efficient medline search tool based on deep syntactic parsing. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, pages 17–20, Morristown, NJ, USA, 2006. Association for Computational Linguistics. 19

Kristen Parton, Kathleen R. McKeown, Bob Coyne, Mona T. Diab, Ralph Grishman, Dilek Hakkani-Tur, Mary Harper, Heng Ji, Wei Yun Ma, Adam Meyers, Sara Stolbach, Ang Sun, Gokhan Tur, Wei Xu, and Sibel Yaman. Who, what, when, where, why?: comparing multiple approaches to the cross-lingual 5w task. In *ACL-IJCNLP '09: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1*, pages 423–431, Morristown, NJ, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-45-9. 24

Jakub Piskorski, Hristo Tanev, and Pinar Oezden Wennerberg. Extracting violent events from on-line news for ontology population. In *BIS'07: Proceedings of the 10th international conference on Business information systems*, pages 287–300, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 978-3-540-72034-8. 26

Dragomir Radev, Timothy Allison, Sasha Blair-goldensohn, John Blitzer, Arda Çelebi, Stanko Dimitrov, Elliott Drabek, Ali Hakim, Wai Lam, Danyu Liu, Jahna Otterbacher, Hong Qi, Horacio Saggion, Simone Teufel, Adam Winkel, and Zhu Zhang. Mead - a platform for multidocument multilingual text summarization. 2004. 23

Adwait Ratnaparkhi. Maximum entropy models for natural language ambiguity resolution. Technical report, 1998. 11

Efstathios Stamatatos, Nikos Fakotakis, and George K. Kokkinakis. Automatic extraction of rules for sentence boundary disambiguation. In *In Proceedings of the Workshop in Machine Learning in Human Language Technology*, pages 88–92, 1999. 15

Fabian M. Suchanek, Georgiana Ifrim, and Gerhard Weikum. Combining linguistic and statistical analysis to extract relations from web documents. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 712–717, New York, NY, USA, 2006. ACM. ISBN 1-59593-339-5. 19

Beth M. Sundheim. Overview of the fourth message understanding evaluation and conference. In *MUC4 '92: Proceedings of the 4th conference on Message understanding*, pages 3–21, Morristown, NJ, USA, 1992. Association for Computational Linguistics. ISBN 1-55860-273-9. 19

Mihai Surdeanu, Sanda Harabagiu, John Williams, and Paul Aarseth. Using predicate-argument structures for information extraction. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 8–15, Morristown, NJ, USA, 2003. Association for Computational Linguistics. 23, 27

Hristo Tanev, Jakub Piskorski, and Martin Atkinson. Real-time news event extraction for global crisis monitoring. In *NLDB '08: Proceedings of the 13th international conference on Natural Language and Information Systems*, pages 207–218, Berlin, Heidelberg, 2008. Springer-Verlag. ISBN 978-3-540-69857-9. 19

David Urbansky, Klemens Muthmann, and Philipp Katz. *TUD Palladian Overview*, December 2010. 37

Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995. ISBN 0-387-94559-8. 13

Wei Wang, Dongyan Zhao, Lei Zou, Dong Wang, and Weiguo Zheng. Extracting 5w1h event semantic elements from chinese online news. In *Web-Age Information Management, 11th International Conference*, pages 644–655, 2010. 27

Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005. ISBN 0120884070. 9

Leixing Xie, Hari Sundaram, and Murray Campbell. Event mining in multimedia streams. In *Proceedings of the IEEE Vol. 96, No. 4*, pages 623–647, April 2008. 24

Sibel Yaman, Dilek Hakkani-Tur, and Gokhan Tur. Combining semantic and syntactic information sources for 5-w question answering. In *In Interspeech*, 2009. 23

Yiming Yang, Tom Pierce, and Jaime Carbonell. A study of retrospective and on-line event detection. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 28–36, New York, NY, USA, 1998. ACM. ISBN 1-58113-015-5. 19

Yiming Yang, Jaime Carbonell, Ralf Brown, Tom Pierce, Brian T. Archibald, and Xin Liu. Learning approaches for detecting and tracking news events. pages 32–43, 1999a. 20

Yiming Yang, Jaime G. Carbonell, Ralf D. Brown, Thomas Pierce, Brian T. Archibald, and Xin Liu. Learning approaches for detecting and tracking news events. *IEEE Intelligent Systems*, 14(4):32–43, 1999b. ISSN 1541-1672. 19

Roman Yangarber. Verification of facts across document boundaries. In *Proc. International Work- shop on Intelligent Information Access*, 2006. 22, 29

Roman Yangarber and Lauri Jokipii. Redundancy-based correction of automatically extracted facts. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 57–64, Morristown, NJ, USA, 2005. Association for Computational Linguistics. 22

Hongyuan Zha. Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '02, pages 113–120, New York, NY, USA, 2002. ACM. ISBN 1-58113-561-0. 22

Jun Zhu, Zaiqing Nie, Xiaojiang Liu, Bo Zhang, and Ji-Rong Wen. Statsnowball: a statistical approach to extracting entity relationships. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pages 101–110, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-487-4. 19

## Statement of authorship

I hereby certify that this diploma thesis has been composed by myself, and describes my own work, unless otherwise acknowledged in the text. All references and verbatim extracts have been quoted, and all sources of information have been specifically acknowledged. It has not been accepted in any previous application for a degree.

Dresden, March 10, 2011