Institut für Informatik, AG Databases and Information Systems

BACHELORARBEIT

Capabilities of Automatic Information Extraction: Evaluation and Comparison with Human Performance

Antje Simon
asimon@inf.fu-berlin.de
Matrikelnummer 3956493

Betreuer: Professor Dr.-Ing. Heinz F. Schweppe, Peter Siniakov

Berlin, 31. Oktober 2006

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides Statt, dass ich die vorliegende Bachelorarbeit bis auf die offizielle Betreuung selbst und ohne fremde Hilfe angefertigt habe und die benutzten Quellen und Hilfsmittel vollständig angegeben sind.

Berlin, den 31. Oktober 2006 .....

Unterschrift

Table of Contents

1. Introduction 5
   1.1 Information Extraction Terminology 6
      Information Extraction 6
      Templates / Target Structure 6
      Training Examples and Expected Results 6
      Performance Measures: Precision, Recall, F-measure 6
   1.2 IE Systems: Architecture and Approaches 8
      1.2.1 ELIE 9
2.Prior Research 11
3. Experiment Procedure 13
      Documents Corpus and Target Structure 13
      Training Examples and Expected Results 13
   3.1 Manual Fact Extraction (Human tester) 14
      Experiment Setup 14
      Training Procedure 14
      Patterns Learned During Training 14
      Observations Made During Training 14
      Fact Extraction Procedure 15
      Observations Made During Fact Extraction 15
   3.2 Automatic Fact Extraction (adaptable IE system ELIE) 16
      The Experiment's Adaptable IE System 16
      Input Data Preparation 16
      Training and Fact Extraction 17
      Observations 17
4. Result Analysis 18
      Evaluation Method 18
      Training Examples / Expected Results 18
      Weaknesses of the Training Examples & Expected Results 23
   4.1 Human Performance Results 24
      Quantitative Analysis – Total Results 24
      Individual Attributes 24
      Number of Tokens Per Extraction 28
      Partial Matches 35
      Duration of training and fact extraction 36
      Qualitative Analysis 37
   4.2 Machine results 39
      Quantitative Analysis: Total Results 39
      Individual Attributes 40
      Number of Tokens Per Extraction 46
      Partial Matches 53
      Operating Times 54
      Qualitative Analysis 56
   4.3 Comparison Human Performance – Machine Performance 58
      Comparison of quantitative evaluation data 58
      Comparison: Operating Times 61
      Comparison: Semantic Extraction Quality 61
5. Conclusion 63
6. Implementation: Evaluation Module for XTract 64
   6.1 New Module Functionality 64
   6.2 Evaluation Module Description 64
References 67

Abstract: As the amount of data provided in natural language texts is constantly growing, it becomes more and more difficult to find the information one is looking for. Collection of the relevant facts in structured form allows easy querying and processing. Machines can process many tasks faster and more cost efficiently. Therefore it is desirable to exploit computer capabilities to solve these problems. In the field of information extraction (IE), systems are developed that identify relevant facts in document, extract, and organize them.
The capabilities of IE systems are usually evaluated by quantitative measures (i.e. how much the results of an IE system agree with those expected). These measures neglect possible usefulness of the incorrect extractions and limitations of extraction quality due to text complexity.
A comparison to human performance can put a system's results into perspective as human beings possess language abilities and domain knowledge that give them an advantage over machines for information extraction tasks. Results achieved by human testers show the limits of what can be expected from an IE system.
We present results of a performance comparison between a human tester and an IE system. The human tester worked under the same conditions as IE systems do, meaning that fact extraction was made based on training and pattern learning from someone else's extractions.
We discovered that human performance was less than mediocre, yet an analysis of the extractions demonstrated that they did contain a very high number of correctly identified facts.
The machine's results were clearly worse than that of the human tester. But there, too, were more useful extractions than the performance measures identified (although less of those than in the human example). The big differences in performance quality for different attributes showed that the complexity of the type of information plays a big role in the quality of results. Inconsistencies in the training examples resulted in bad performance for the machine and the human tester. This suggests that the quality of the training examples and expected results also influence what can be achieved.
In order to evaluate the performances, we developed an evaluation module for the fact extraction tool XTract.

1. Introduction

Information and knowledge are often made available in form of unstructured natural language texts, e.g. newspaper articles and scientific publications. With the number of accessible texts growing, relevant information on an area of interest cannot be efficiently collected and organized by reading all related documents. It is desirable to exploit computer capabilities to solve these problems, as machines can accomplish many tasks faster and more cost efficiently than people if the amount of data is large enough to outweigh the time and money spent on development and maintenance. Yet in order to develop techniques that perform a task, it is necessary to formalize it. Natural language is very rich and complex; it contains cultural and subjective aspects in addition to objective facts, and there are many different ways to express a piece of information in an unstructured natural language text. This makes it difficult to find algorithmic approaches that filter facts out of unstructured natural language texts.

Unlike basic arithmetic calculations, there is no simple algorithm in information extraction that reliably delivers the correct extractions, due to the complexity of human language. IE systems use intricate algorithms that produce approximations. Therefore performance measures are needed that determine the quality of the results.

It is tempting to consider only the quantitative measures as they can be calculated automatically and are easy to compare. Yet there are more factors to consider in order to get a complete picture of the level of data these systems can provide. A closer look at the task complexity and a comparison with human performance gives more insight into the quality of the results a system delivers [8].

In depth extraction analysis can point out weaknesses of the algorithm that help future developments. Possible patterns of an IE system's results can also determine for which tasks the current version can be utilized.

Adaptive information extraction systems are designed to handle different types of document corpora, which vary in difficulty. While texts for one task have a simple structure, directly stated facts and frequently recurring key words (e.g. job advertisements), another task requires the extraction of information from complex texts containing difficult-to-process linguistic properties (e.g. indirectly stated facts, metaphors). Processing a difficult corpus cannot be expected to produce the same high quality as the results of a simpler task.

Thus the performance measures a system achieved on one set of documents with unknown difficulty do not predict how well the same system will do on another corpus.

People have better natural language processing abilities than machines. Although text comprehension is dependent on a person's language understanding and domain knowledge, the majority of a report's audience (e.g. for a research paper: experts in the domain) should be able to grasp the given information. Performance quality achieved by human testers can help determine the difficulty of a corpus and show limitations of what kind and quality of extracted information can be expected from complex texts. IE systems' extractions with similar results can be considered high quality, even if the absolute numbers are mediocre or low.

In this thesis, we compared machine to human performance in an experiment with both working under the same conditions. We determined IE limitations and examined quality differences, providing a differentiated picture of the capabilities of the information extraction system.

In the implementation part of this thesis, we added an evaluation module to the fact extraction tool XTract that provides the calculation of performance measures and viewing of extractions. We used this tool for the evaluation of the results of our experiment.

1.1 Information Extraction Terminology

Information Extraction

When requesting data on a specific topic within a given collection of texts, *Information Retrieval (IR)* is an instrument to find documents of a given corpus that are likely to contain information about a topic of interest. However, the number of returned texts can be too large to analyze and process manually.

The area of *Information Extraction (IE)* applies techniques and algorithms that find relevant data and store it in a structured form that allows easy querying and automatic processing. IE tasks include Named Entity recognition (finding and classifying pieces of information), co-reference resolution (identifying entities that refer to the same object), determining relationships between entities and assignment of the entities to appropriate target structure attributes [11].

When handling extracted data, information extraction requires less analysis than information retrieval, and the structured form makes it easier to process the data. But it is more difficult to implement and the accuracy can depend on the studied domain (i.e. the subject matter of the texts).

Templates / Target Structure

Templates define the structure of the extracted data. Templates, also called *target structures*, contain slots for the different attributes of requested data (e.g. event or object). Each event (e.g. terroristic act, seminar, job announcement, etc.) or object is described by several entities (e.g. date, location, weapon, position; components, size, capacity) that relate to the event or object. These are the event/object attributes. The entities are extracted and stored in a template; each of them in the slot that describes the relationship it has to the event (e.g. occurred_on, is_held_at, is_job_position_title; has_this_type_of_componentA, length).

The template should contain all the information (as attributes) that is needed for the expected queries and future processing. The creation of an appropriate target structure is not trivial. It requires knowledge of the domain how the relevant information is given in the texts.

Approaches for fact extraction require a given target structure. Some information extraction approaches generate templates automatically. Although this reduces human effort and knowledge necessary to adapt to a new domain, the resulting target structures may not be as appropriate as those constructed manually, i.e. they do not contain attributes that are needed for future processing.

Training Examples and Expected Results

For the evaluation of extractions produced by an IE system (or test person), the actual facts included in the corpus documents must be known. These facts are usually defined and extracted by an expert in the domain (or sometimes more than one) [4]. Part of these extractions that are given to the machine for training, thus they are referred to as *training examples*. The performance measures are calculated by comparing the machine's / tester's results for the documents that were not part of training to the rest of the prepared extractions. Those are the *expected results*. There are usually several machine runs executed, with different parts of the data provided for training (i.e. each run has different parts of the extractions as training examples and as expected results). That is why there is no previously defined split of *training examples* and *expected results* and all prepared extractions can be referred to as *training examples*.

Performance Measures: Precision, Recall, F-measure

For the evaluation of extractions of a text corpus, the templates filled by the IE system and the training examples are compared. Each slot fill is either *correct* (also *true positive, TP*; i.e. matching the expectation exactly), *partially correct* (partially matching the expectation, containing more or less tokens), or *incorrect* (*false positive, FP*; i.e. not matching a training example slot). If a

fact in a training example slot is not included in the extractions, it is considered *missing* (*false negative, FN*). For more detailed performance measures (which are not considered in our study), additional categories are defined as well [6].

Precision, recall, and F-measure, which are the standard metrics in IR, are the most common measures in IE as well.

Precision looks at how many of the made extractions are correct (TP/(TP+FP)).

Recall describes how many of the training examples are included in the made extractions (TP/(TP+FN)).

Although it is not common, partially correct matches are sometimes included in the calculations of precision and recall. (e.g. weighed with ½).

F-measure is the harmonic mean of precision and recall, weighing precision and recall equally or differently, depending on their importance for the current task. As the goal in IE is to reach both high recall and high precision (not just a high result for one and a low result for the other), the harmonic mean is used for the F-measure because extreme values (one high and one low value) result in a lower F-measure than two medium values.

Evaluating extractions with *one answer per occurrence* considers extractions to be correct if only if the extracted facts and the position of them in the texts match the expected result. *One answer per document* expects that each template slot contains one fill per document and counts an extraction to be a true positive if the fact matches the expectation, even if it was extracted from a different part of the document.

The human and machine performances in our experiment were calculated with the new evaluation module of XTract and ELIE's scorer. The calculations used one answer per occurrence evaluation and did not count partial matches as correct matches (they are in fact both FP – extractions that are not in the expected result, and FN – the expected extraction was not made). We used the F_1 F-measure, which weighs precision and recall equally.

1.2 IE Systems: Architecture and Approaches

Information extraction tasks are very complex and different approaches have been and are being built resulting in a diverse range of existing systems. In general, there are three main tasks: input text preprocessing, learning, and fact extraction.

During preprocessing, a linguistic analysis of the text is made. For example, the text is split into different sentences, part-of-speech tags are created, named entities are identified, and multiple references to the same object (e.g. "The seminar starts at 10. It ends at 12." - "seminar" and "it" refer to the same thing) are connected (co-reference resolution). The extraction model receives domain and corpus specific parameters during the learning phase. Most IE systems analyze annotated texts to create extraction rules or patterns. Sometimes the learning process requires user supervision, e.g. the first extractions are rated by the user as correct or incorrect and the system learns from this input. Then the extraction model is applied to the texts from which information is to be extracted. Facts are identified and stored in the appropriate template slots, creating structured output.

The generic IE system description by J. Hobbs [9] identifies ten modules performing different steps of the information extraction process. Most IE systems perform the steps of the generic systems, although they may be arranged differently. The generic system allows easy characterization of an IE system as developers can describe their system in terms of differences and implementations of their system compared to the generic system.

According to Hobbs, the generic IE system consists of the following modules that are arranged in a pipeline architecture (i.e. each module receives the output from the previous module as input):

1. Text Zoner: turns a text into a set of segments
2. Preprocessor: adds part-of-speech (POS) tags to the lexical items of the text segments
3. Filter: removes irrelevant sentences
4. Preparser: identifies small-scale structures in sequences of lexical items
5. Parser: produces parse tree fragments
6. Fragment Combiner: tries to combine parse tree or logical form fragments for whole sentences
7. Semantic Interpreter: generates semantic structure
8. Lexical Disambiguation: solves ambiguities, resulting in unambiguous predicates
9. Co-reference Resolution: links all occurrences of the same entity
10. Template Generator: fills the templates

One way of classifying existing IE systems is based on the learning techniques, models, and resources, although many systems use a combination of approaches.

Siniakov and Siefkes [1] distinguish three main classes of adaptive system approaches: rule learning, knowledge-based, and statistical.

As the name suggests, ***rule learning approaches*** employ techniques that induce rules from given resources, such as annotated texts ("training corpus"), domain information, keyword lists, POS tags, or supervision by users.
The subclass *automatic pattern and template creation* requires syntactic and semantic domain information and uses some statistical methods for the generation of templates and patterns. Only little user interaction is needed.
Covering algorithms are a type of inductive learning. A fully annotated training corpus is expected as input. Specific rules are generalized in order to cover as many positive examples as possible.
Relational rule learners are based on covering algorithms but look at positional and other relations between features and not only at predefined feature-combinations, as covering algorithms do.

The assumption that word concept is dependent on the context is the basis for *case-based approaches*. These systems build a case base from the training corpus which contains word features for every word. When processing a text, the context features of each word are identified and similar entries in the case base are used to determine what to extract.
Systems using *wrapper induction* can process automatically generated structured and semi-structured texts, exploiting the regularities in the text. The Boosted Wrapper Induction system is a development making the wrapper induction approach capable of processing unstructured texts. It builds patterns consisting of token sequences that identify facts. Since there are many ways to express a fact in free text, this approach does not produce high recall.
IE^2, a *hybrid approach*, works with handwritten rules but also handles the complex tasks of template unification and event merging beyond sentence level.

Instead of a fixed given target structure, *Horn clauses* have been suggested for the structured representation of information in a text in one of the ***knowledge-based approaches***. As this approach extracts all information from a text, it is related to IE. No knowledge sources are given as input but constant user interaction is required to build a set of cases which consist of dependencies between the verb and its arguments in a clause.
Ontology-based extraction is another knowledge-based approach. It utilizes an ontology of the domain and relevant items are identified by applying regular expressions. Since no new regular expressions are learned, this approach is currently not flexible enough for free texts but can process pre-formatted text elements such as enumerations.
Systems implementing the *thesaurus-based extraction* approach require several dictionaries and a thesaurus. User input is needed to assign the correct word sense to a word and build semantic networks for the training texts. Based on the networks, rules are created and generalized, which are then applied to the text to be processed.

Probabilistic parsing is a ***statistical approach***. It holds a syntax model from a large corpus that is applied to the training corpus with semantic tags. This results in a parse tree with semantic and syntactic annotations. Categories and POS tags are assigned based on statistical methods.
Hidden Markov models are the basis for several other statistical approaches, so are *Bayesian networks*.
Token classification approaches treat information extraction as a token classification task, classifying every document position as either a start tag of an extraction, an end tag, or neither. ELIE, an IE system using a classification algorithm, will be described in detail in the following chapter.

1.2.1 ELIE

ELIE is an adaptive IE system developed by Finn and Kushmerick [2,3].

ELIE uses token classification to accomplish information extraction. All positions of a document are classified as being either the start of a field to be extracted, the end of a field to be extracted, or neither. Two levels of learning (L1 and L2) are employed, with Level One producing high precision and Level Two improving recall while lowering precision only slightly.

During preprocessing, input files with POS and chunking information for each token are generated using Brill's Part-Of-Speech tagger. An instance consists of a token (e.g. word, number, punctuation mark in a document) together with its POS and chunking information, reference to the list in the gazetteer (user-defined dictionary of lists of first names, last names, countries, cities, etc) which contains this token (if applicable), orthographic information, such es upper-case, alphabetic, or punctuation, and information about the surrounding instances.

For Level One learning, each instance of the training examples is classified as a positive or negative example of a start or an end tag. ELIE currently uses Weka's SMO support vector machine implementation as its learning algorithm. This produces models for start and end tags with many

negative instances (all tokens in the document that are not a start/end tag) but only a small number of positive instances. This leads to a low probability for predicting a tag, yet the predictions are likely to be correct, yielding high precision but low recall. The tag-matcher matches predicted start and end tags looking at how many tokens are between them and how many occurrences of start/end-tag-pairs in the training examples had the same number of tokens between them.

The start model for the Level Two learner is built with only those instances of the document that occur in a fixed distance before an end tag (for the end model: fixed distance after a start tag). This results in models with a higher number of boundary instances, i.e. instances that are positive examples of start or end tags. Thus L2 learning is called convergent boundary classification. The L2 model is likely to produce higher recall but lower precision than L1. If the L2 model was applied to the an entire document, many false positives would be generated, worsening performance. That is why the L2 model is only applied to regions where the L1 model has made a prediction. This way, L2 improves recall while keeping L1's precision high.

A. Finn and N. Kushmerick conducted experiments [3] on the seminar announcement dataset, which contains e-mails announcing university seminars; they used a window length of 3, L2 lookahead/lookback of 10 tokens, 50/50 train/test splits, single-slot-occurrence evaluation (i.e. at most one slot fill per attribute in a document; all co-references of a expected result are considered correct). They compared ELIE's results to those published for BWI, LP^2, and RAPIER for similar settings:

	$ELIE_{L1}$			$ELIE_{L2}$			BWI			LP^2				Rapier	
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
speaker	95.8	76.2	84.9	91.0	86.0	**88.5**	79.1	59.2	67.7	87.0	70.0	77.6	80.9	39.4	53.0
location	96.1	75.9	84.8	93.1	80.7	**86.5**	85.4	69.6	76.7	87.0	66.0	75.1	91.0	60.5	72.7
stime	99.0	94.4	96.6	98.6	98.5	98.5	99.6	99.6	**99.6**	99.0	99.0	99.0	96.5	95.3	95.9
etime	99.0	77.8	87.0	95.7	97.3	**96.4**	94.4	94.4	94.4	94.0	97.0	95.5	95.8	96.6	96.2

Comparing ELIE to other IE systems, seminar announcement corpus. From [3]

The table above shows that ELIE's performance is almost equal to or better than the performance of the other IE systems. It also shows that ELIE's L2 classification produces higher recall than L1 alone and worsens precision only slightly, resulting in better F_1.

In another experiment, Finn and Kushmerick ran the jobs dataset (containing job advertisements). Here they used all-slot-occurrences evaluation, meaning that all occurrences of a field need to be extracted to be counted as a correct. A comparison to other IE systems' results was made, but it is not clear what evaluation methods the other systems used.

	$ELIE_{L1}$			$ELIE_{L2}$			LP^2			Rapier		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
id	100	99.5	99.7	100	99.5	99.7	100.0	100.0	**100.0**	98.0	97.0	97.5
title	69.3	40.1	50.7	57.2	54.6	**55.8**	54.0	37.0	43.9	67.0	29.0	40.5
company	94.2	69.6	**79.8**	90.1	71.5	79.5	79.0	76.0	71.9	76.0	64.8	70.0
salary	73.9	52.8	61.4	71.5	62.1	66.3	77.0	53.0	62.8	89.2	54.2	**67.4**
recruiter	88.3	75.6	81.3	87.0	77.7	**82.0**	87.0	75.0	80.6	87.7	56.0	68.4
state	93.7	92.2	**92.9**	92.4	93.1	92.7	80.0	90.0	84.7	93.5	87.1	90.2
city	96.0	93.2	94.6	95.2	94.9	**95.1**	92.0	94.0	93.0	97.4	84.3	90.4
country	97.7	93.9	95.8	97.4	94.3	**95.8**	70.0	96.0	81.0	92.2	94.2	93.2
language	93.6	87.2	90.2	93.3	89.5	**91.4**	92.0	90.0	91.0	95.3	71.6	80.6
platform	87.1	64.6	74.1	84.8	75.5	79.8	81.0	80.0	**80.5**	92.2	59.7	72.5
application	85.5	51.2	63.9	81.0	61.4	69.7	86.0	72.0	**78.0**	87.5	57.4	69.3
area	72.5	29.0	41.4	61.8	40.3	48.7	70.0	64.0	**66.9**	66.6	31.1	42.4
req_years_ex	89.0	69.0	77.5	80.8	79.6	**80.0**	79.0	61.0	68.8	80.7	57.5	67.1
des_years_ex	96.5	70.0	80.8	93.1	75.4	82.9	67.0	55.0	60.4	94.6	81.4	**87.5**
req_degree	90.2	65.3	75.0	84.8	75.0	79.0	90.0	80.0	**84.7**	88.0	75.9	81.5
des_degree	95.5	45.1	58.9	67.8	50.8	55.2	90.0	51.0	65.1	86.7	61.9	**72.2**
post date	95.2	99.0	97.0	95.2	100.0	97.5	99.0	100.0	99.5	99.3	99.7	99.5

While LP^2 and Rapier achieved better scores for some attributes, ELIE's F-Measure on those attributes was still close to 50% or higher, showing that ELIE's performance was good for this corpus as well.

2. Prior Research

Comparison of Human and Machine IE Performance

Craig A. Will published the results of his comparison between human and machine IE performance in 1993 [4]. He looked at the extractions of machines that participated in the MUC-5/Tipster evaluation (English microelectronics corpus) and those of expert coders during the different steps of the development of the training examples for the MUC-5 evaluation.

An analysis of the corpus data indicated that information extraction from the MUC-5/Tipster corpus is a more difficult task than making extractions from the MUC-3 corpus, which we use in our experiment [12].

After the templates for the MUC-5/Tipster evaluation had been developed by experienced analysts, a set of detailed extraction rules was defined. Four expert analysts were given the task of preparing the training examples for the corpus. Each analyst made extractions from all texts although only half of these extractions were used in the development of the training examples. A fifth analyst prepared training examples for another set of texts.

Will looked at the percent error (i.e. error per response fill as calculated by the SAIC scoring program) of the extractions made by the analysts compared to the training examples that were not created using the same analyst's extractions and those developed by the fifth analyst (which correspond better to the extractions produced by machines than considering extractions which were used in the training examples development) and presented the following findings.

The average percent error was 33.2% (scores of the individual analysts: 32%, 32%, 33%, 39%) when comparing an analyst's extractions to training examples that were developed by two other analysts; in comparison to the training examples made by the fifth analyst, the average percent error was 28.3% (22%, 26%, 28%, 37%). The three best IE systems in the MUC-5/Tipster evaluation reached percent error scores of 62%, 63%, and 68% -- The error rate of the machines was twice as high as that of the human analysts. It was noted, however, that some of the systems tested were incomplete, thus results of released systems may be better.

Another interesting finding in this study was that the human error rate was so high. Even extractions that were made by an analyst who later used his extractions and those of another analyst to produce the training examples had an error rate of 15.5% (the other set of extractions used for the examples reached 27% error). This shows that there is no fixed "correct extraction" that experts agree on independently of each other and analysts change their extractions when presented with another expert's opinion. Not surprisingly, analysts are biased towards their own extractions.

Comparison of Human and Machine Performance: Classification of News Story Leads

In the summer of 2003, King and Lowe published the results of a study comparing the extractions made by an IE tool with those of human coders [5].

They tested the Virtual Research Associates, Inc Reader, which processes the first sentences ("leads") of Reuters Business Briefing news stories. This was developed based on the common practice that the first sentence summarizes the key points of the events reported in a news story. The Reader not only identifies events but also assigns them the code of appropriate categories from the Integrated Data for Events Analysis (IDEA) and WEIS ontologies. The human coders were three undergraduate students that had no text coding experience (their knowledge about the domain is not mentioned). The texts used in the evaluation were about the collapse of Yugoslavia and the conflict in Bosnia. Expert human coders decided on the correct classification codes for each lead.

The results were presented in different ways, applying different weighing functions for the categories and looking at both the top level ("aggregate") IDEA categories as well as at the detailed classification categories. Depending on the calculation method, the proportion of correct codes ranged from 23% (categories weighed equally, detailed categories) to 70% (frequent categories weighing more, aggregate categories) for the human coders. In all cases, the machine made at least as many correct classifications as the human coder with the least number of correct codes. When looking at the proportion of correctly assigned aggregate categories, the machine scored almost as well or even better than the best human coder.

They also looked at whether events were correctly identified as events (no matter whether they were coded correctly) and non-events were correctly identified as non-events. The machine classified 93% of the events as events; the best human coders between 80% and 94%. The only task the machine performed significantly worse at than the human coders was the correct identification of non-events. While the human coders recognized non-events as non-events 92% to 100% of the time, the machine did so only 23% of the time, thus assigning classification codes to leads that contained no events. Since the number of news stories containing no events was rather small and the coded non-events did not appear in some categories more than in others, these incorrect assignments had no big impact on the extraction results.

In conclusion, King and Lowe found that human and machine performance were almost identical for the classification task in this setup.

3. Experiment Procedure

For the comparison of human and machine performances, we had a human tester and an adaptive information extraction system perform the same information extraction task. The testing conditions were as similar as possible.

The human tester and the machine both received the same information extraction task. Their results were subsequently evaluated and a comparison between their performance was made. The human tester's extractions were evaluated by the evaluation module in XTract (see ch. 6. Implementation: Evaluation Module for XTract), which was also used for the qualitative analysis of the extractions of both the human tester and the IE system. The IE system's own scorer calculated the performance measures for its results.

Documents Corpus and Target Structure

The text corpus for our experiment consisted of 650 MUC-3 (terroristic act) documents. These documents included news stories and transcripts of radio/television broadcasts and interviews. Thus the texts varied in difficulty, style, and amount of information covered. While news stories put facts directly, the transcribed speeches contain more phrases and metaphors [7].

The target structure had been prepared prior to the experiment. The template for a terroristic act could hold one extraction per attribute. It contained the following attributes:

- `time`
- `attackdate`
- `perpetrator`
- `perpetrator_number`
- `perpetrator_organisation`
- `action`
- `weapon`
- `victim_target`
- `victim_others_animate`
- `victim_others_inanimate`
- `location` *(first version of the target structure)*
- `town`, `department`, `country` *(second version of the target structure)*

The `location` attribute turned out to be insufficient as most of the documents identify the location of a terroristic act by more than one piece of information. It received bad results during test evaluations. That is why the `location` attribute was replaced by the more detailed `town`, `department`, and `country`. The extractions for these attributes were subsequently added to the database. The experiment followed the same sequence: First information for the non-place attributes and `location` was extracted; the `town`, `department`, and `country` slots were filled in a second processing of the documents.

Training Examples and Expected Results

A database containing the extractions for training and testing had been created as well. The extractions were made by a graduate student working in the field of information extractions with non-expert knowledge of the domain. They mirrored his understanding of what the relevant facts were. No extraction rules had been defined before starting this procedure. This resulted in a database that included the information an unbiased person considers important.

The corpus was divided into a training batch and a test batch, each consisting of half (325) of the documents. The training batch was used to learn from the examples what kinds of extractions are expected. The human tester was to find patterns in the training data that the creator of the training/test extractions (unknowingly or knowingly) used. The tester was to extract facts in the test batch applying these patterns. The IE system used the training data to build a data model which was used to make predictions on the test documents.

3.1 Manual Fact Extraction (Human tester)

Experiment Setup

For the comparison of the human tester's performance to the IE system's, the manual fact extraction's working conditions were made as similar to the machine's as possible. While it is impossible to produce the exact same conditions for both (because of human knowledge, etc), we tried to imitate the machine's working conditions as much as possible.

We took the following measures to achieve this:

- Training on half of the MUC-3 documents. This comprised reviewing the training extractions for these document and noting recurring patterns
- No special training in the domain
- No rules or patterns for fact extraction were provided
- Application of the patterns learned in the training stage training during fact extraction of the test corpus (the other half of the MUC-3 documents)

Training Procedure

The tester started training without any special domain knowledge or experience in information extraction, and received no data resources or instructions in addition to the training examples.

The training was carried out by viewing the training documents and the corresponding examples in XTract. XTract is a fact extraction tool that displays documents, highlights the extractions in the texts, and shows all corresponding extracted terroristic acts (attribute slots and extracted strings for each) in a separate window [10]. Recognized patterns were written down.

The training phase was carried out "part-time" on four consecutive days.

Patterns Learned During Training

Patterns learned during training:

- Extractions did not overlap (exception: `location` extractions overlapped with extractions for the specific geographic attributes because these they were not part of the same target structure). *Note*: Overlapping extractions are not common in information extraction.
- A person's name was extracted completely (first and last names). Explanations/details following the name (e.g. “Enrico Gonzales, *mayor of...*”, “Enrico Gonzales, *head of the XYZ organization*”) were usually not included.
- Other victims that were wounded (not killed) were often not extracted.
- Texts that consisted of interviews or statements of people from different organizations or parties were treated like regular articles.
- Some extracted terroristic acts contained only little information and were summaries of several acts that had been committed in the past months/years.
- Sometimes one terroristic act was extracted twice if it was described in detail in different paragraphs of an article, both extracted acts consisting only of the attributes mentioned in one paragraph.

Observations Made During Training

Every person has some level of previous knowledge of the domain. Most people agree on and know simple facts (e.g. "Bogota is the capital of Colombia"). But human knowledge has a larger range; it contains more than mere facts. The level of knowledge differs from person to

person, depending on experience, education, even preferences and opinions about the relevance of pieces of information, and it is prone to error.

Human beings do not pay attention to every detail or make a mental list of all of their observations when they read a text. Different people pay attention to different aspects or properties of words/sentences/reports and may develop fact extraction patterns according to their focus. It is easier to find a pattern in annotations if one would apply the same properties to differentiate between possible extractions. For example, if one person would extract the first occurrence or the most detailed description of a fact, they would probably notice if these patterns are present or if the last or least detailed occurrences were extracted instead. But unexpected patterns (such as the -unlikely- extraction of the description with the most letters) are difficult to recognize if one does not pay attention to the number of characters.

As people start training with their own ideas on which parts of a report they consider relevant, they may tend to classify someone else's annotations in relation to their own patterns ("the extractions for the weapon attribute are the ones I would have made").

If there are several extractions that are inconsistent with an otherwise correctly applied pattern (such as: the pattern is not used 25% of the time) may be difficult for a person to recognize the pattern.

Fact Extraction Procedure

The fact extraction functionality of XTract [10] was used for manual fact extraction. The extractions were stored in a relational database.

The extraction phase started a few days after the training had been completed. Each of the test documents was opened in XTract. The reports were read and then the tester decided which parts to extract by applying the patterns learned during training and copied the extractions into a database. Attributes belonging to the same terroristic act were part of one relation in the database. Each relation contained only one slot for each attribute, which meant that the best fit for each attribute in a terroristic act had to be selected.

The attributes `town`, `department`, and `country` were added after all extractions had been made because the creator of the training examples followed the same procedure.

After the extraction process had been completed, the extractions in the database were not altered.

Limitations: The used version of XTract was unable to extract facts containing an apostrophe or more than 200 characters (entry limit of the database attributes). Therefore some facts could only be extracted partially into the appropriate slots.

Observations Made During Fact Extraction

In our experiment, only one string could be inserted into a slot. As some facts were given in more than one part of a text, it was necessary to decide which piece of information to extract. Predicting which of these facts would be included in the expected results was sometimes difficult as no pattern for cases like this had been learned. In addition to that, this resulted in extractions that did not contain all the information that the tester considered to be relevant.

Many of the texts give the date and location of the reported attack in relation to the publishing date and place of the report (e.g. "yesterday", "last month", "on Monday", "in this city"). When looking at the extractions alone without the complete report, these facts do not provide much information.

In certain domains and report types, like that of this corpus (terroristic acts; including transcripts of speeches), unconfirmed information is often given in a report, such as potential perpetrators or a person's opinion. Extractions of unconfirmed information cannot be distinguished from confirmed facts when looking at them alone without the document itself. Although it would

make sense to include only confirmed information in the database, it might be desirable to include unconfirmed pieces of information if no other text about this event exists in the corpus. The training examples contained facts extracted from speech and interview transcripts.

3.2 Automatic Fact Extraction (adaptable IE system ELIE)

The Experiment's Adaptable IE System

We used the IE system ELIE to represent information extraction systems for our comparison between machine and human performance. We picked ELIE because it is an up-to-date IE system producing the same high quality as other up-to-date systems, making it a representative for today's IE capabilities (see ch. 1.2.1 ELIE, page 9).

ELIE uses a token classification approach (see ch. 1.2.1 ELIE). ELIE can process only one attribute at a time. Therefore it does not extract terroristic acts (i.e. slot fills for different attributes that belong to one act) but identifies only occurrences of the given attribute input.

The ELIE installation includes its own scorer that evaluates the extractions, calculating the number of true positives, false positives, and false negatives. We used this scorer to obtain ELIE's performance measures.

Input Data Preparation

As ELIE could not process the annotated files or database entries that existed for the corpus, the documents were preprocessed.

ELIE provides a preprocessor that accesses Brill's Part-Of-Speech tagger. The annotated documents were given as input. The preprocessor split the texts into tokens (e.g. word, number, punctuation mark) and added part-of-speech tags, target structure annotations (if applicable) and suffix information for each token. The ELIE installation contains lists of stopwords (i.e. frequently used words that contain no relevant information if looked at them without context, such as "the", "always", "with", "someone" as well as common verbs like "say", "use", etc.), first names and last names. The ELIE preprocessor added appropriate tags to tokens that appeared in one of these lists.

In order to evaluate ELIE's general performance on this corpus ten different shuffles were generated at random. Each shuffle (called "split" in ELIE) lists the corpus documents in a different order. ELIE accesses these lists and uses the first documents on the list for training (in that order) and the others for fact extraction ("testing"). With the different shuffles, ELIE used different documents for training and making extractions in each run. Running ELIE with all shuffles provided a good visualization of the quality and deviation of ELIE's performance on the experiment data.

In addition to ELIE's performance on the regular corpus, we tested its performance on a pre-classified corpus, from which we had removed the documents without extractions in the training examples. This was due to the fact that when giving an IE system a pre-classifed corpus which contains only documents that contain extractions, the performance often improves. ELIE's algorithm classifies every token in the texts as a start field of an extraction, an end field of an extraction, or neither. This results in many tokens classified as "neither" and a comparatively small number of starts and ends. The pre-classification increases the portion of starts/ends. This should lead to a higher recall for ELIE's L1 learner in particular because more starts and ends in the training examples mean a higher probability for predicting a start/end.

This pre-classified corpus contained 389 documents. A second set of 10 shuffles was produced that contained only these documents.

Training and Fact Extraction

We set up ELIE to train on half of the preprocessed documents of each shuffle and make extractions from the second half just like the human tester did. The performance measures were calculated with ELIE's own scorer.

A template attribute and the shuffle information (i.e. document order lists) were given as input and ELIE performed training and fact extraction for all shuffles consecutively in one run. The token IDs of the extractions and expected results were returned as output. Then the scorer calculated the performance measures based on these results.

The runs of ELIE with the pre-classified shuffles were started afterwards.

The ten regular shuffles included the one that the human tester used for manual extraction. The qualitative analysis and performance comparison was based on the results of this shuffle.

Observations

Unlike the procedure the creator of the training examples and the human test person followed, ELIE extracted only one attribute at a time in one run and did not access the extractions it had produced for the other attributes. This resulted in cases of overlapping extractions for different attributes. In addition to that, ELIE produced overlapping extractions for the same attribute as well.

4. Result Analysis

We evaluated the performance of the human tester and the machine separately first, looking at the differences between the made extractions and the expected results. Then we compared the findings of both.

Evaluation Method

We calculated the performances using one answer per occurrence extraction evaluation. Partial matches did not count as matches (they are in fact both FP – extractions that are not in the expected result, and FN – the expected extraction was not made). Our F-measure values weighed precision and recall equally ("F_1").

We had a closer look at the number of tokens of the extractions and dependencies between token number and performance quality as well as the partial matches.

In addition to the quantitative evaluation, a qualitative analysis was performed in order to identify error patterns.

Training Examples / Expected Results

The training examples / expected results consisted of extractions made for all 650 documents of the MUC-3 terroristic act corpus.

As we ran ELIE with ten different shuffles to look at its average performance, part of the training example extractions for one shuffle were expected results a second shuffle and vice versa. Thus a clear differentiation between training examples and expected results (and their properties) cannot be made for the machine performance but averages for number of extractions, etc. were determined instead.

Table 1) Expected results: Number of extractions per attribute

Attribute	***Extractions***
action	650
attackdate	285
country	153
department	108
location	341
perpetrator	348
perpetrator_number	72
perpetrator_organisation	175
time	72
town	308
victim_others_animate	175
victim_others_inanimate	84
victim_target	649
weapon	246

The table and plot below show the token lengths of the expected result extractions broken down into the different attributes based on all 650 documents of the corpus. As the corpus was split into 325 training and 325 extraction documents, the expected results (i.e. extractions from the those documents of the corpus that the test person/IE system made extractions for) for a shuffle contain *on average* one half of these extractions. The exact numbers for the shuffle used for the qualitative analysis of the human and machine performances are given in the analysis chapters.

Table 2) Expected results: Number of tokens per extraction

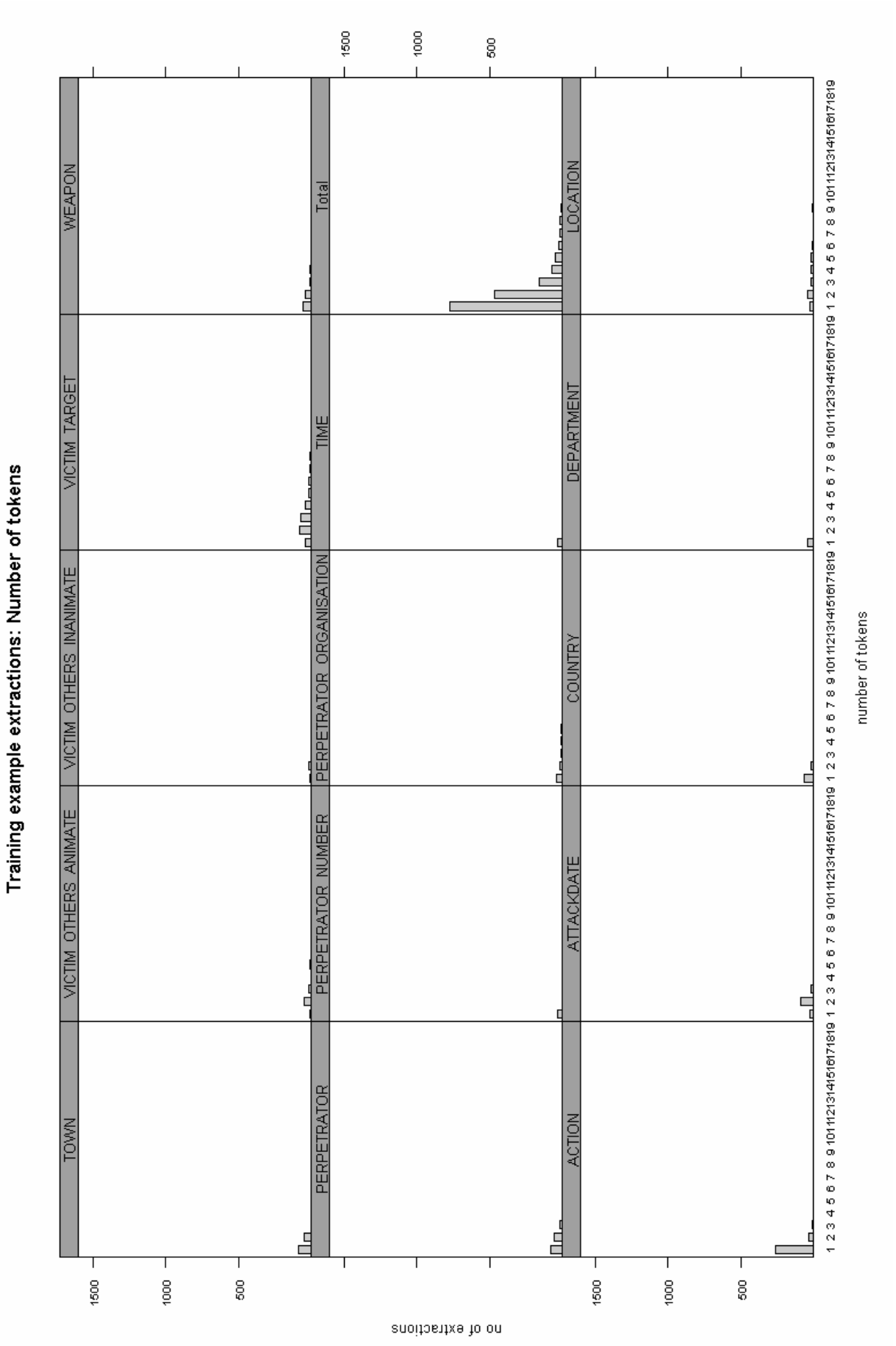
No of tokens	Attribute	No of extr.
1	total[1]	1628
2	total[1]	1032
3	total[1]	393
4	total[1]	186
5	total[1]	153
6	total[1]	74
7	total[1]	52
8	total[1]	47
9	total[1]	33
10	total[1]	14
11	total[1]	14
12	total[1]	4
13	total[1]	6
14	total[1]	2
15	total[1]	8
16	total[1]	6
17	total[1]	4
18	total[1]	1
19	total[1]	3
20	total[1]	4
21	total[1]	1
23	total[1]	1
1	action	545
2	action	68
3	action	27
4	action	6
5	action	3
10	action	1
1	attackdate	65
2	attackdate	176
3	attackdate	39
4	attackdate	1
5	attackdate	1
6	attackdate	1
7	attackdate	1
8	attackdate	1
1	country	121
2	country	32
1	department	80
2	department	19
3	department	8
4	department	1
1	location	53
2	location	89
3	location	50
4	location	28
5	location	34
6	location	25
7	location	13
8	location	13
9	location	16
10	location	5
11	location	2
12	location	2
13	location	1

No of tokens	Attribute	No of extr
1	perpetrator_number	66
2	perpetrator_number	5
6	perpetrator_number	1
1	perpetrator_organisation	68
2	perpetrator_organisation	35
3	perpetrator_organisation	15
4	perpetrator_organisation	16
5	perpetrator_organisation	35
6	perpetrator_organisation	1
7	perpetrator_organisation	2
8	perpetrator_organisation	2
9	perpetrator_organisation	1
1	time	60
2	time	6
3	time	1
4	time	4
6	time	1
1	town	196
2	town	97
3	town	9
4	town	6
1	victim_others_animate	12
2	victim_others_animate	88
3	victim_others_animate	23
4	victim_others_animate	13
5	victim_others_animate	11
6	victim_others_animate	6
7	victim_others_animate	7
8	victim_others_animate	7
9	victim_others_animate	4
11	victim_others_animate	2
16	victim_others_animate	2
1	victim_others_inanimate	17
2	victim_others_inanimate	40
3	victim_others_inanimate	4
4	victim_others_inanimate	5
5	victim_others_inanimate	6
6	victim_others_inanimate	2
8	victim_others_inanimate	3
9	victim_others_inanimate	4
10	victim_others_inanimate	1
11	victim_others_inanimate	1
20	victim_others_inanimate	1
1	victim_target	68
2	victim_target	175
3	victim_target	159
4	victim_target	79
5	victim_target	54
6	victim_target	32
7	victim_target	24
8	victim_target	16
9	victim_target	7
10	victim_target	7
11	victim_target	9
12	victim_target	2

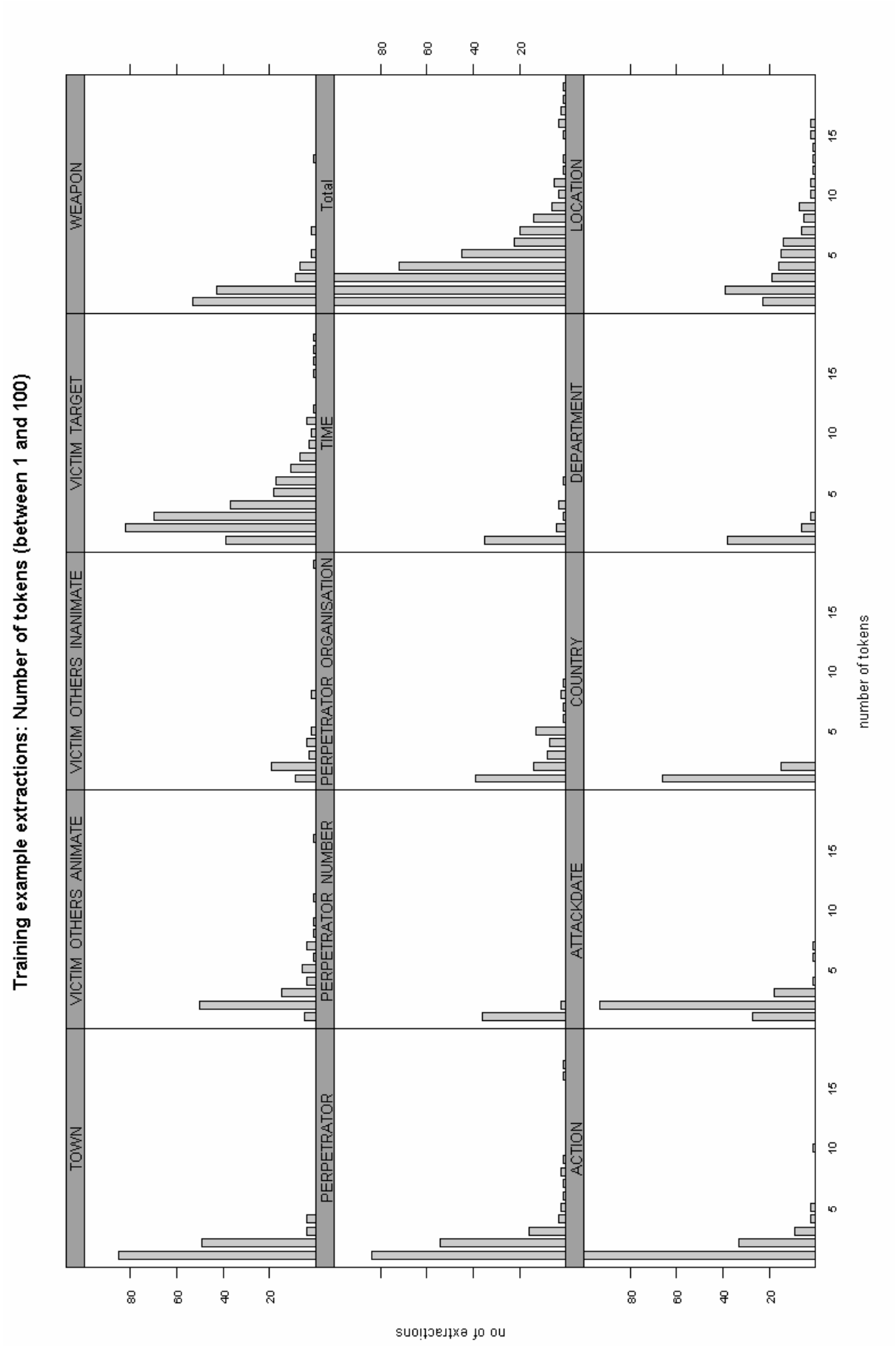
No of tokens	Attribute	No of extr.
14	location	1
15	location	6
16	location	2
17	location	1
1	perpetrator	153
2	perpetrator	124
3	perpetrator	40
4	perpetrator	11
5	perpetrator	5
6	perpetrator	5
7	perpetrator	3
8	perpetrator	2
9	perpetrator	1
13	perpetrator	1
16	perpetrator	1
17	perpetrator	1
21	perpetrator	1
13	victim_target	3
14	victim_target	1
15	victim_target	2
16	victim_target	1
17	victim_target	2
18	victim_target	1
19	victim_target	3
20	victim_target	3
23	victim_target	1
1	weapon	124
2	weapon	78
3	weapon	18
4	weapon	16
5	weapon	4
7	weapon	2
8	weapon	3
13	weapon	1

[1] the total results refer to the extractions for all atributes excluding the location attribute

Plot 1) Expected Results: Distribution of extractions per number of tokens (by attributes) overview



Plot 2) Expected Results: Distribution of extractions per number of tokens (by attributes)
Detailed look at extraction frequencies between 1 and 100



The training example extractions usually contain less than five tokens. It is easy to see that attributes such as `action`, `attackdate`, and `country` are commonly described by just a few words ("assassination", "terrorist attack"; "July 17, 1989", "today"; "Colombia", "El Salvador"). In contrast, `victim_target` information can be rather long as some reports give victim information in form of lists of several names of victims or other detailed information.

Weaknesses of the Training Examples & Expected Results

The training examples and expected results were considered to be the actual facts in the reports for our evaluation of human and machine performances. However, we determined inconsistencies and errors in these extractions.

Errors and inconsistencies noticed in the training examples:

- The name of the perpetrators' organizations were sometimes not extracted.
- Announcements/threats were extracted as facts.
- The number of perpetrators was sometimes included in the `perpetrator` attribute.
- In phrases like "bomb explosion", the `weapon` was sometimes included in the `action`
- Some towns were extracted as countries and vice versa
- In a few documents, the town named in the beginning of the article (stating place and date of publication) was extracted
- Inconsistent use of the geographic attributes, such as extracting a city as a location and not as a city
- Extraction of towns/countries in the article that did not describe the terroristic act's location

The following errors were detected in the expected results:

- Several extracted countries and towns were not given as location information for the terroristic act they were assigned to.
  In *"Oqueli Colindres was kidnapped as he was heading to the airport in Guatemala City to take a flight to Nicaragua"* (026), "*Nicaragua*" was put in the country slot. "*California*" was extracted as the country in a report about "*approximately 50 Salvadoran leftist activists demanded the mediation of Costa Rican president Oscar Arias as a condition to leave the Costa Rican embassy which they forcefully occupied on the morning of 3 October. [...]",* which states that *"President Arias is currently in California*." (547)
  Some of these extractions are not directly mentioned as the location of the act but correct considering the context, such as "*Peru*" in : "*Political terrorism has become part of everyday life in Peru. [...] In Tirapata village (Puno department) a Sendero Luminoso detachment seized the local mayor and shot him [...]*" (650).
- "*The Columbian city of Medellin"* (451) was extracted as location in the old version of the target structure but neither Columbia nor Medellin were extracted as country or town, respectively.
- There were a few erroneous extractions of towns as countries or districts and neighborhoods as departments. In document 524, "*Bogota*" was extracted in two different places: once as a town and once as a country. The names of "*San Miguelito district*" (621) and "*Miraville neighborhood*" (622) were included as departments.
- "*22 March*" in "*[...]since the military actions began on 22 March*" (093) was included as an expected result for location.

4.1 Human Performance Results

Quantitative Analysis – Total Results

With a recall of a little less than one half and precision of roughly one third, the human performance was worse than expected. Even for "simple" attributes the F-measure was only around 50%. The following direct comparison of the tester's extractions with the expected results gives more details about the properties these results and gives explanations.

Table 3) Human performance: total performance measures

	Recall	*Precision*	*Partial*	*F_1-Measure*
Total (with location attribute)*	46.8 %	35.0 %	10.6 %	40.1 %
Total (with town,department, country)*	47.2 %	36.5 %	8.6 %	41.2 %

* The performance measures were calculated twice: once using the general attribute "location", and once using the more specific attributes "town", "department", "country"

The human tester extracted more terroristic acts than the expected results contained (acts: 442 vs 326; attributes: 2084 vs 1620 respectively, see below). Because of this, even if all of the expected extractions were included in the made extractions (i.e. recall = 100%), the precision would still be less than 78%. The expectations included extractions from 11 documents from which the tester did not extract anything. These facts were obviously not recalled by the tester. Furthermore the test person extracted more terroristic acts from some documents than the expected results (most notably 12 vs 7 for 638, 5 vs 2 for 639, 5 vs 0 for 295, 6 vs 2 for 142) and this happened less often the other way around. These additional acts also reduce the human performance's recall.

Number of extracted terroristic acts	
- in the training example	326
- in the human tester's extractions	442
Number of documents	
- total	325
- for which the training examples contained at least one terroristic act:	189
- for which the training examples contained at least one act but the human results not:	11
- from which the human tester extracted at least one act:	236
- from which the human tester extracted at least one act but the training example not:	58

Individual Attributes

After learning patterns from the training extractions and using general (but not expert) knowledge of the domain, we expected the human tester's extractions to have rather high precision and recall for easy to identify attributes that are described by only one or two tokens, such as `time`, `date`, `perpetrator_ number`, `perpetrator_organisation`, `town`, `department`, `country`. Especially after not finding many patterns in the training examples, our expectations for the other attributes were lower.

The recall of about 65% and precision of 40% for `attackdate` and `time`, and a recall of around 50% for `perpetrator_organisation` were satisfying in this respect, although not as good as expected. The tester achieved the best results with the `weapon` attribute (recall: 60% , precision: ~50%); which was surprising because we had assumed the `attackdate` and `time` extractions to produce the best performance.

Another surprise was the rather good recall of `action`, `victim_target`, and

victim_others_animate. These attributes were difficult to extract, due to the fact that these facts were often repeated in a report, were split into different sentences in the documents, or contained several tokens. The varying and unexpectedly low quality of the performance for town, department, and country is discussed below.

The number of partial matches is especially high for location and the three victim information slots, attributes whose entries contain token numbers varying from only one to many (see Number of Tokens Per Extraction, page 28). The number of partial matches show that the expected results and the tester's identified the same parts of the document as the description of the facts for these slots.

In reports about terroristic acts, many facts (especially perpetrator and victim information) are often given multiple times. Without using a pattern such as "always extract the first occurrence", it was not surprising that the tester sometimes extracted a different occurrence of these pieces of information than what was expected. The better performance of the "one answer per document" evaluation supports this assumption. Recurring appearances of a fact in a text sometimes contain different words, like pronouns or synonyms, because repetition is considered bad style. In some cases repetitions of facts have some words in common (e.g. first occurrence: title, first name, last name; second occurrence: only last name), other times they do not (e.g. if pronouns are used). Our evaluation algorithm could not recognize co-references. This allowed the assumption that the extractions in the training examples and the tester's referred to the same persons/items more often than the performance measures suggest.

The total results for the target structure including the location attribute were slightly worse than those for the target structure containing the more specific attributes town, department, and country. We expected the detailed attributes to receive much better results than location because location is a very general attribute and many reports gave more than one piece of location information (e.g. street, part of town, town). The provided location information often contained several expressions in different parts of the text. This made it difficult for the tester to decide which of these pieces and how many of the tokens to put into the location slot, leading to fewer matches with the expectations. In contrast, town, department and country should be very easy to identify because they are limited to geographic words and the names of these places have only one or very few tokens. This theory holds true for town and department. There were a few cases of geographic descriptions that are more difficult to classify into the specific attributes, such as "on the highway between towns C and D", "on the D-country border to country E". Most documents, however, name no more than one town, department, and country when reporting where a terroristic act took place. Therefore we expected the performance for town, department and country to be very good.

There are two common factors that limit the performance for simple attributes like this. First of all, limited geographical knowledge may cause mistakes and omissions. Secondly, multiple occurrences of a town/department/country name in a report may lead to picking the a different one than the expected results hold. For the latter case the "one occurrence per document" performance would show that the extracted place matches that in the training examples.

The human performance for the town attribute, while still not very good (F-measure one answer per occurrence: 41.4%, one answer per document: 58.1%) was much better than the performance for location (F-measure for one answer per occurrence: 15.4%, one answer per document: 21.4%). This proved our theory that the specific geographic attributes would yield better measures than location alone. The same held true for the department attribute, although it received lower scores (F-measure for one answer per occurrence: 28.8%, one answer per document: 36.6%). However, this can be explained by the fact that the tester could not always tell if a geographic name is that of a part of town, town, region, department or other, and the expectations contained errors as well.

The names of the countries in South America, however, were well known to the test person and the creator of the expected results. Thus the fact that country received by far the worst results

(F-measure for one answer per occurrence: 7.5%, one answer per document: 10.8%) was very unexpected. It may seem like a mistake at first, but a closer look at the extractions could explain this phenomenon at least in part:

* The human tester made only 25 `country` extractions (training example: 81), making it impossible to have a recall of over one third. In several documents the the country names were simply overlooked, which could only be be explained by a lack of concentration, as the country names were very easy to identify in texts.

* Another factor that caused low recall were errors and inconsistencies in the training examples – including extractions of countries that were mentioned in a document but not as the location of the reported terroristic act and a few extractions of town names as countries and vice versa. The errors had no pattern (not all country names appearing in a text were extracted; town names that were extracted as countries only once or twice and were correctly extracted as towns in other documents); that is why the tester did not make the same erroneous extractions unless the same errors were made incidentally.

Table 4) Human performance: Performance measures ("one answer per occurrence" evaluation)

attribute	***expected***[1]	***made***[2]	***recall***	***precision***	***partial***	***F_1-measure***
action	307	376	50.2 %	41.0 %	3.7 %	45.1 %
attackdate	141	229	65.2 %	40.2 %	5.7 %	49.8 %
perpetrator	166	212	48.2 %	37.8 %	6.1 %	42.3 %
perpetrator_number	38	38	36.8 %	36.9 %	7.9 %	36.8 %
perpetrator_organisation	86	138	52.3 %	32.7 %	9.4 %	40.2 %
time	44	76	65.9 %	38.2 %	2.6 %	48.3 %
victim_others_animate	89	134	42.7 %	28.4 %	15.7 %	34.1 %
victim_others_inanimate	40	66	32.5 %	19.7 %	10.6 %	24.5 %
victim_target	295	393	44.4 %	33.3%	16.0 %	38.1 %
weapon	117	133	59.8 %	52.7 %	12.8 %	56.0 %
town	142	177	46.5 %	37.3 %	6.2 %	41.4 %
department	46	59	32.6 %	25.4 %	0	28.6 %
country	81	25	4.9 %	16.0 %	0	7.5 %
location	160	182	16.8 %	14.3 %	24.2 %	15.4 %
total (with location)	1494	2000	46.8 %	35.0 %	10.6 %	40.1 %
total (with town,department, country)	1620	2084	47.2 %	36.5 %	8.6 %	41.2 %

[1]number of expected extractions [2]number of slots filled in the human tester's extractions

As we mentioned above, the performance measures for "one answer per document" were better than those for "one answer per occurrence" due to the fact that there were multiple extractions of co-references that were identical to the expected results.

Table 5) Human performance: Performance measures ("one answer per document" evaluation)

Attribute	***Recall***	***Precision***	***Partial***	***F_1-Measure***
action	66.9 %	56.7 %	6.9 %	61.3 %
attackdate	75.0 %	50.9 %	6.3 %	60.7 %
perpetrator	58.0 %	45.7 %	7.9 %	51.1 %
perpetrator_number	42.9 %	40.5 %	8.1 %	41.7 %
perpetrator_organisation	62.3 %	35.8 %	15.1 %	45.5 %
time	74.4 %	46.8 %	1.6 %	57.4 %
victim_others_animate	43.8 %	32.0 %	18.0 %	37.0 %
victim_others_inanimate	28.9 %	20.0 %	14.5 %	23.7 %
victim_target	56.4 %	44.5 %	23.8 %	49.8 %
weapon	64.2 %	57.5 %	14.2 %	60.7 %
town	63.6 %	53.5 %	7.9 %	58.1 %
department	33.3 %	40.6 %	0	36.6 %
country	7.4 %	20.0 %	0	10.8 %
location	22.8 %	20.2 %	29.5 %	21.4 %
Total (with location)	56.1 %	43.4 %	14.1 %	49.0 %
Total (with town,department, country)	56.4 %	46.0 %	11.5 %	50.7 %

Number of Tokens Per Extraction

The number of tokens per extractions for each attribute (see Expected Results: Distribution of extractions per number of tokens (by attributes) overview, page 21) demonstrated that the extracted facts for most of the attributes usually contained less than five tokens. While common token number is a pattern of IE extractions and systems such as ELIE apply this pattern to their predictions, the human tester did not pay attention to token numbers as a pattern while learning from the training extractions.

The following table and plots show the token number distribution of the human tester's extractions. For a comparison to the expected results, see Table 2, Plots 1 and 2.

Table 6) Human performance: Number of token per extraction by attribute

number of tokens	attribute	number of extractions	expected results[2]	difference
1	total[1]	858	776	82
2	total[1]	581	464	117
3	total[1]	238	155	83
4	total[1]	100	72	28
5	total[1]	60	45	15
6	total[1]	27	22	5
7	total[1]	32	20	12
8	total[1]	27	14	13
9	total[1]	13	6	7
10	total[1]	7	3	4
11	total[1]	11	5	6
12	total[1]	5	1	4
13	total[1]	0	1	-1
15	total[1]	0	1	-1
16	total[1]	4	3	1
17	total[1]	4	2	2
19	total[1]	1	1	0
20	total[1]	1	1	0
1	action	282	260	22
2	action	52	33	19
3	action	29	9	20
4	action	6	2	4
5	action	3	2	1
10	action	1	1	0
1	attackdate	59	27	32
2	attackdate	130	93	37
3	attackdate	24	18	6
4	attackdate	7	1	6
6	attackdate	3	1	2
7	attackdate	1	1	0
1	country	22	66	-44
2	country	3	15	-12
1	department	46	38	8
2	department	10	6	4
3	department	1	2	-1
1	location	17	23	-6
2	location	34	39	-5
3	location	42	19	23
4	location	19	16	3
5	location	21	15	6
6	location	15	14	1
7	location	7	6	1
8	location	4	5	-1

number of tokens	*attribute*	*number of extractions*	*expected results*[2]	*difference*
9	location	6	7	-1
10	location	4	2	2
11	location	5	2	3
12	location	3	1	2
13	location	0	1	-1
14	location	2	1	1
15	location	1	2	-1
16	location	1	2	-1
1	perpetrator	91	84	7
2	perpetrator	71	54	17
3	perpetrator	27	16	11
4	perpetrator	5	3	2
5	perpetrator	4	2	2
6	perpetrator	2	1	1
7	perpetrator	4	1	3
8	perpetrator	2	2	0
9	perpetrator	0	1	-1
16	perpetrator	1	1	0
17	perpetrator	1	1	0
1	perpetrator_number	30	36	-6
2	perpetrator_number	5	2	3
1	perpetrator_organisation	55	39	16
2	perpetrator_organisation	31	14	17
3	perpetrator_organisation	11	8	3
4	perpetrator_organisation	10	7	3
5	perpetrator_organisation	13	13	0
6	perpetrator_organisation	5	1	4
7	perpetrator_organisation	1	1	0
8	perpetrator_organisation	6	2	4
9	perpetrator_organisation	0	1	-1
1	time	60	35	25
2	time	5	4	1
3	time	6	1	5
4	time	3	3	0
6	time	0	1	-1
1	town	93	85	8
2	town	66	49	17
3	town	8	4	4
4	town	5	4	1
1	victim_others_animate	7	5	2
2	victim_others_animate	45	50	-5
3	victim_others_animate	22	15	7
4	victim_others_animate	9	4	5
5	victim_others_animate	9	6	3
6	victim_others_animate	2	1	1
7	victim_others_animate	9	4	5
8	victim_others_animate	8	1	7
9	victim_others_animate	1	1	0
11	victim_others_animate	3	1	2
16	victim_others_animate	3	1	2
1	victim_others_inanimate	6	9	-3
2	victim_others_inanimate	19	19	0
3	victim_others_inanimate	10	3	7
4	victim_others_inanimate	5	4	1
5	victim_others_inanimate	1	2	-1
8	victim_others_inanimate	1	2	-1

number of tokens	attribute	number of extractions	expected results[2]	difference
20	victim_others_inanimate	1	1	0
1	victim_target	30	39	-9
2	victim_target	113	82	31
3	victim_target	89	70	19
4	victim_target	42	37	5
5	victim_target	29	18	11
6	victim_target	15	17	-2
7	victim_target	17	11	6
8	victim_target	10	7	3
9	victim_target	12	3	9
10	victim_target	6	2	4
11	victim_target	8	4	4
12	victim_target	5	1	4
15	victim_target	0	1	-1
16	victim_target	0	1	-1
17	victim_target	3	1	2
19	victim_target	1	1	0
1	weapon	77	53	24
2	weapon	31	43	-12
3	weapon	11	9	2
4	weapon	8	7	1
5	weapon	1	2	-1
7	weapon	0	2	-2
13	weapon	0	1	-1

[1] the total results refer to the extractions for all attributes excluding the location attribute

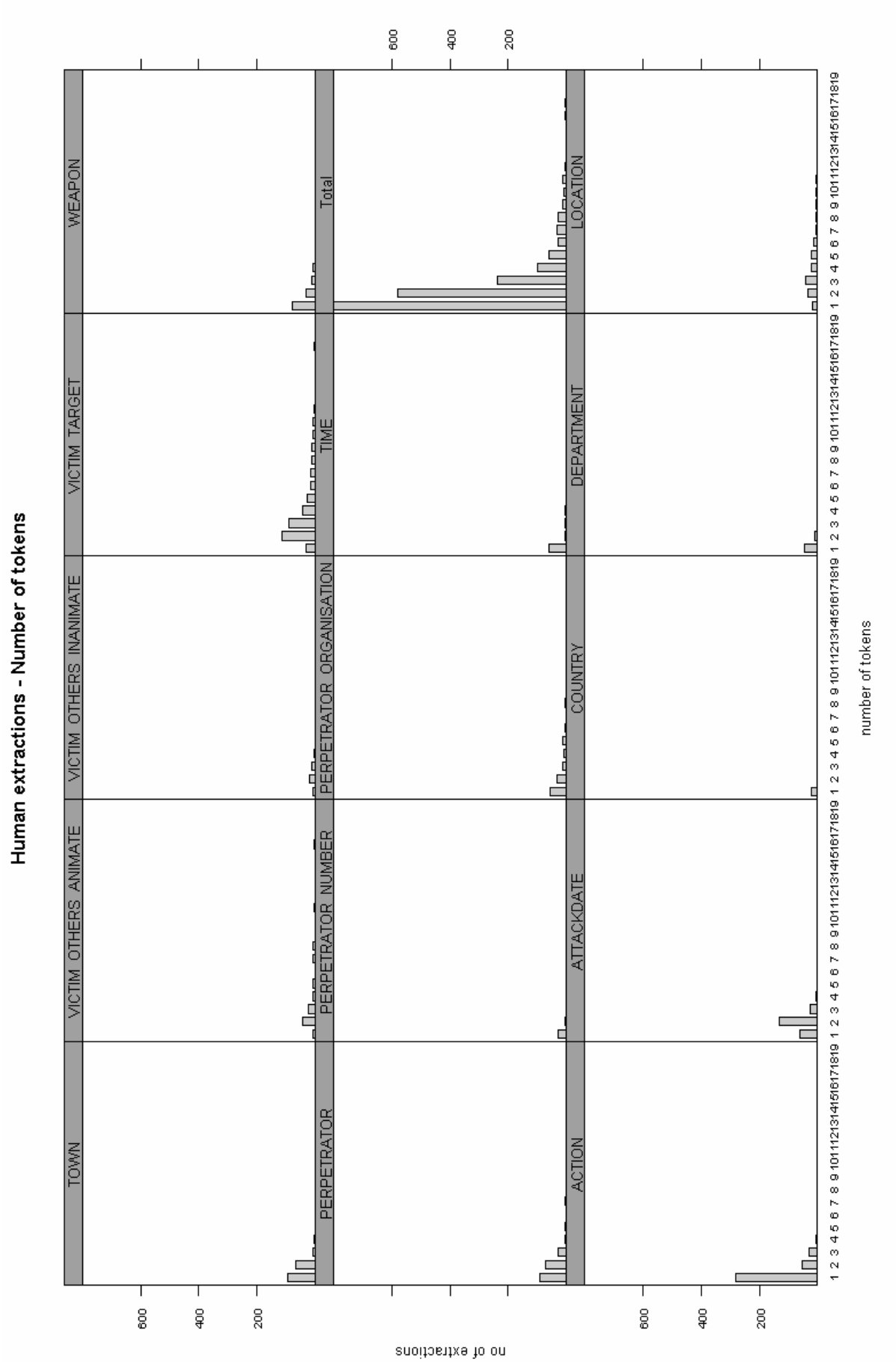
[2] the frequency of token number for the expected results refers to the same documents that the extractions were made for; they are not an average based on the number of extractions for the entire corpus from the previous chapter

While there were a number of big differences between the number of made and expected extractions – in most cases more extractions were made than expected – the distributions of token numbers are similar.

This observation is visualized very well in the plots below and those for the training examples (Plots 1,2). Please note that the plot for the expected results is based on the entire corpus (650 documents) and the human performance looks at the extractions of only 325 documents. The comparison of the plots was made under the assumption that the distribution of token numbers of a random shuffle is similar to that of the whole corpus. For easier visual comparison, the y-axes of the plots also differ by a factor of two.

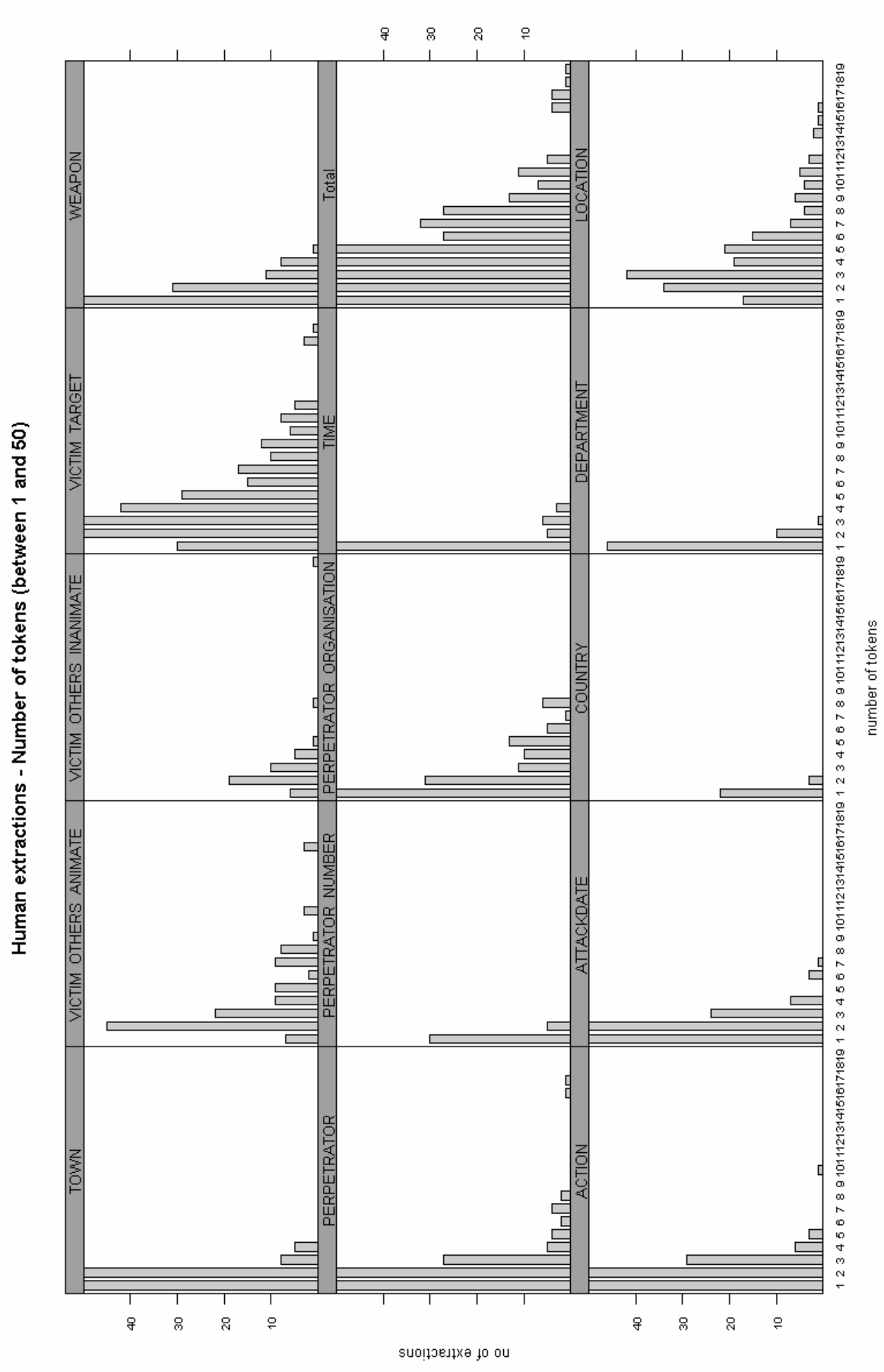
Although the common number of tokens for an attribute was not a learned pattern, the understanding that the information for some attributes is given in few words while there can be long expressions for others was part of the previous language knowledge that the creator of the expected results and the test person apparently shared.

Plot 3) Human performance: Distribution of extractions per number of tokens (by attributes) overview



Plot 4) Human performance: Distribution of extractions per number of tokens (by attributes)

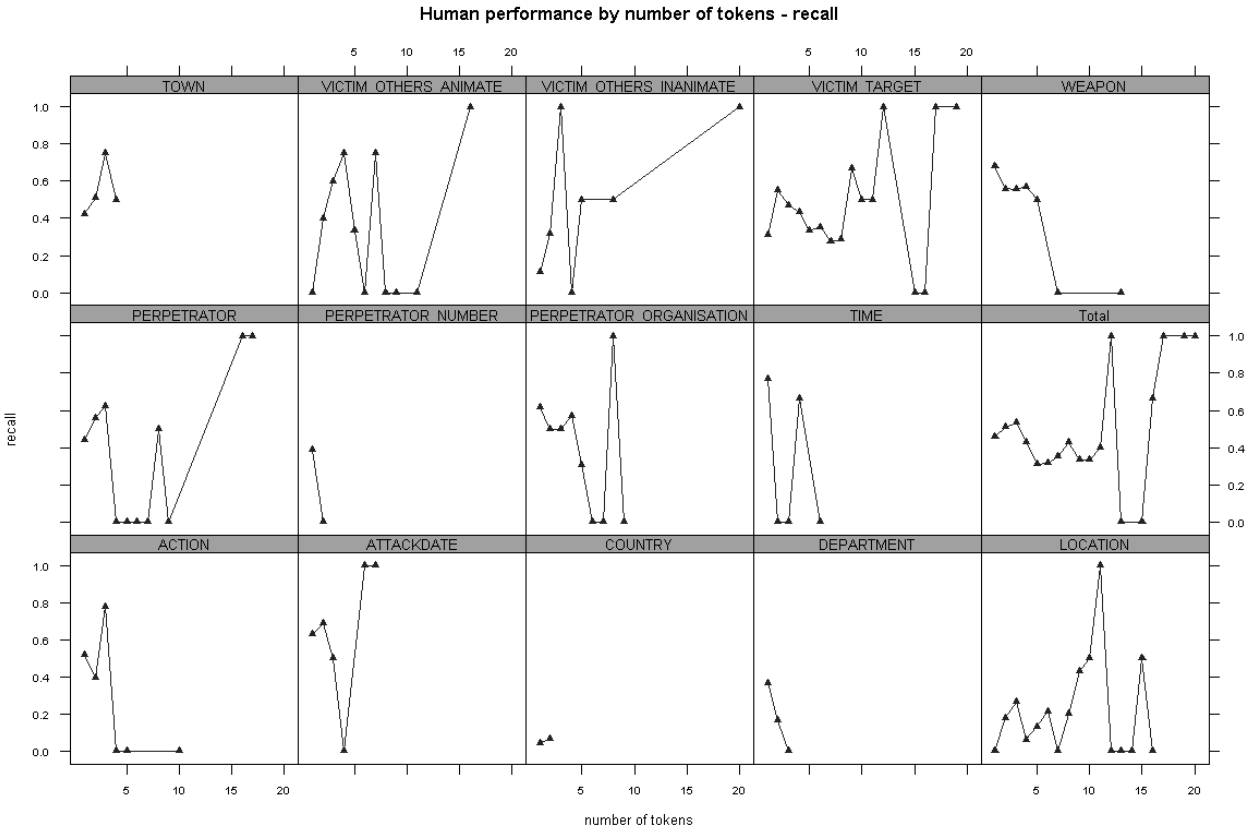
Detailed look at extraction frequencies between 1 and 50



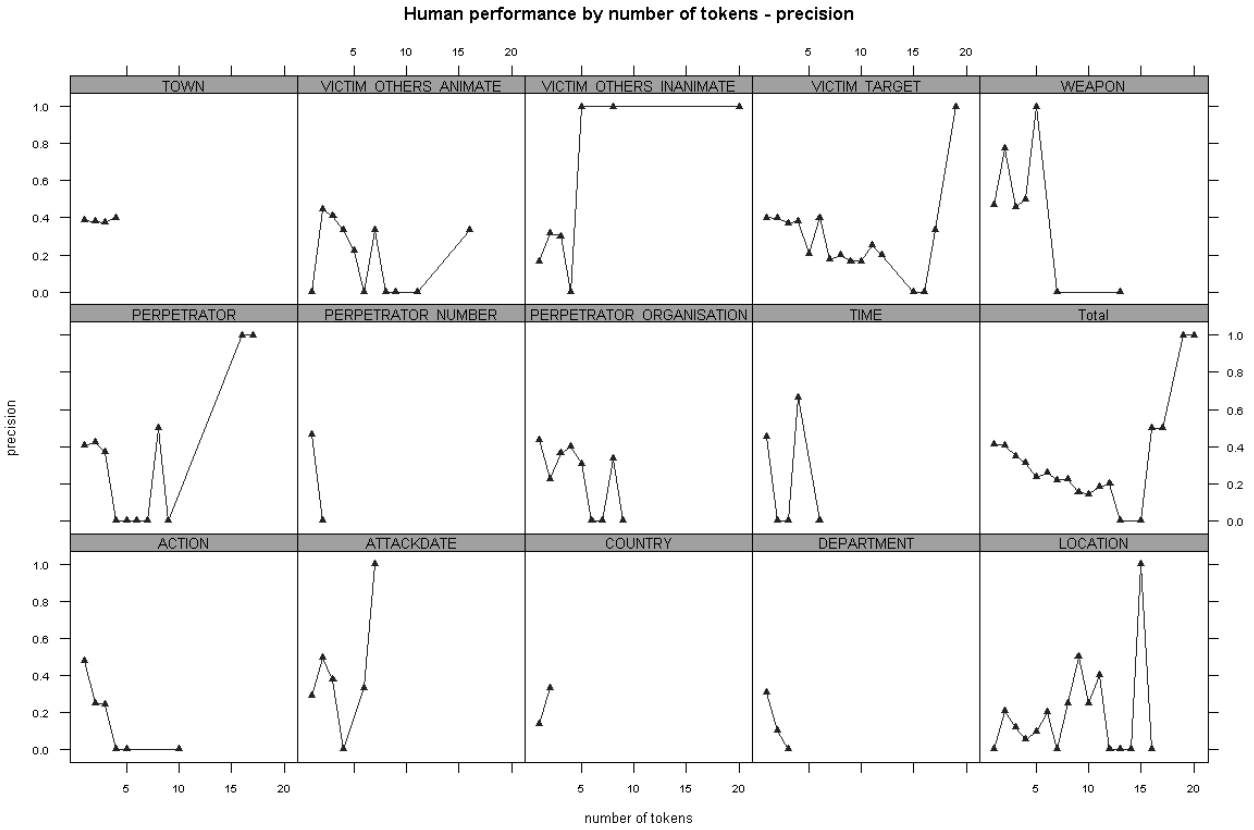
While it sounds logical that recall and precision would better for facts expressed with only one or few tokens – they tend to give precise information (e.g. "assassination", "terrorist attack"; "July 17, 1989", "today"; "Colombia", "El Salvador") and are less likely to have ambiguous starts/ends, which long descriptions of facts are likely to have – the performance measures broken down into number of tokens did not support this theory. In fact, looking at the plots 5 and 6 alone suggests that recall and precision tended to be highest for extractions that are made up of many tokens. This phenomenon can be explained by the fact that there were only few extractions having many tokens. This made it impossible to draw conclusions about patterns for the performance for long extractions.

Since most extractions contained less than five tokens, we had enough data to make observations about the performance of those. The best values were often achieved by extractions holding more than one token (but less than five). Although our initial theory was that short extractions would reach the best results, we identified a reason for the better measures achieved by long extractions. Extractions with many tokens usually contained more detail (e.g. the full names of people, dates including day, month, year, etc.). If those facts were repeated, the repetition was shorter. The more detailed occurrence seemed to be the better source of information. This was a useful selection pattern, and thus was more likely to be extracted. If all occurrences were short or there was only one, this common pattern did not apply.

Plot 5) Human performance: Recall by number of tokens (by attribute)



Plot 6) Human performance: Precision by number of tokens (by attribute)



Partial Matches

As noted above, the similar previous knowledge and understanding of the reports by non-expert human readers in general and specifically of the two people whose extractions we looked at, often resulted in the identification of the semantically same expressions as the relevant facts to be extracted. However, there were often different views about where a piece of information started and ended. This explains why the human results contained a large number of partial matches.

In the analysis, we differentiated between three different kinds of partial matches:

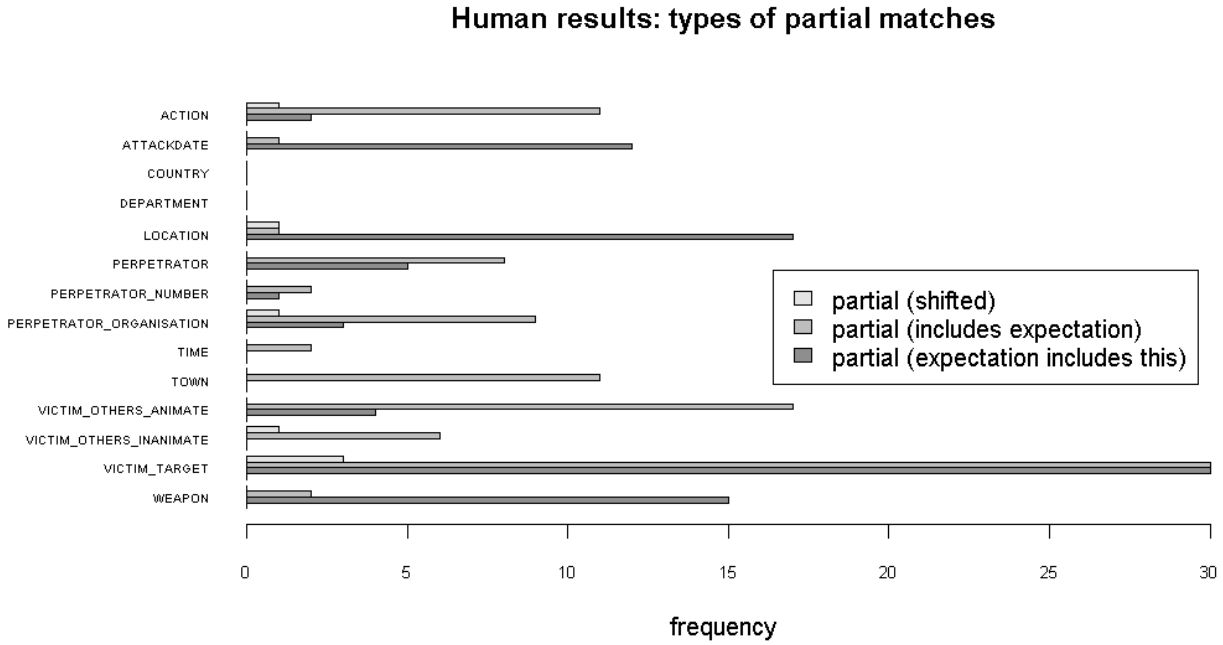
- made extraction is included in the expectation (i.e. later/same start and earlier/same end but not a match) – put simply: "less extracted"
- made extraction includes the expectation (i.e. earlier/same start and later/same end but not a match) – "more extracted"
- made extraction is shifted compared to the expectation (i.e. starts and ends earlier or starts and ends later; both contain at least one token that the other does not)

Table 7) Human extractions: Types of partial matches

Attribute	***Included in expectation***	***Includes expectation***	***Shifted***	***Percentage of extractions***
action	2	11	1	3.7%
attackdate	12	1	0	5.7%
country	0	0	0	0
department	0	0	0	0
location	17	1	1	24.2%
perpetrator	5	8	0	6.1%
perpetrator_number	1	2	0	7.9%
perpetrator_organisation	3	9	1	9.4%
time	0	2	0	2.6%
town	0	11	0	6.2%
victim_others_animate	4	17	0	15.7%
victim_others_inanimate	0	6	1	10.6%
victim_target	30	30	3	16.0%
weapon	15	2	0	12.8%
total	89	100	7	8.6%

Number of partials with the same start as the expectation	64
Number of partials with the same end as the expectation	100

Plot 7) Human results: Types of partial matches



Although the overall results of the total and attribute analyses indicated that the human tester tended to extract more than what was expected, the differentiation of partial matches of all attributes combined showed that there were only a few extractions more of the partials that included the expected results than of those that were included in the expectations.

When looking at each attribute individually, however, there was always one of these two types of partials that was much more frequent than the other, with the exception of `victim_target` (same number of both types), `location` and `department` (no partials).

Since many of the the reports contained not only multiple facts applicable to the `location` attribute but also long location descriptions that left more than one option for reasonable starts and ends of the extractions, the fraction of partial matches was high (24%) for this attribute.

Duration of training and fact extraction

We did not record the amount of time spent on training and fact extraction. Both tasks were carried out during a time period of a few consecutive weeks, yet no minimum or average hour-per-day value was calculated. The slow network connection to the databases with training and made extractions increased the duration time invested in this part of the experiment.

Qualitative Analysis

In general, the made extractions were semantically correct (i.e. correct part-of-speech; weapon extractions were words describing weapons and so on) and most of the time taken from descriptions of terroristic acts. There were only few extractions that made no sense as fills for the corresponding attribute.

Common errors in the human extractions

Apart from regular errors, i.e. overlooking facts or extracting the wrong thing due to ignorance of the relevant facts (mostly concerning geographic names), we found a few recurring extraction patterns that did not match the expectations:

- The tester did not extract 12 acts that were extracted in the expected results. Apart from three of these, they were left out because the tester did not consider them to be terroristic acts. Most of them did not describe a specific incident but summarized several acts of the past years (e.g. "*[...] More than 200 officials have been murdered since the Colombian justice system began its struggle against the drug lords in 1981 [...]"* (document 397)). Others were not the main topic of the report.

It may be debatable if summaries of terroristic acts which provide only little information should be included in the extraction database. However, as the training examples included these, the tester should have learned the pattern that general event descriptions are to be extracted. The tester did indeed extract some of these acts (some of which were not included in the examples); e.g. *"[...] in the past few months, the FMLN has used so-called mobile armed platforms against the principal army garrisons in San Salvador.[...]"* (044), *"[...] asked about the money collected in the kidnapping of impresario Jorge Born, Roberto Perdia said that 'this matter was not discussed' with president Menem [...]"* (599)). Thus this pattern was followed with but with exceptions.

- A couple documents reported only threats which the tester extracted as acts (e.g. *"[...] security at the Colombian embassy in Kingston has been tightened following a bomb threat on Monday [...]"* (511), *"[...] a radio station in the capital received a phone call in which an unidentified person threatened a generalized attack on radio stations throughout the country. [...]"* (544)).

The corpus contained transcripts of statements from guerrilla groups. These included announcements of their future activities.

- *"[...] the general command decrees a national transportation stoppage beginning at dawn on 31 May [...]"* (211) is an example of threats that the tester extracted.
- The expectation was to extract "terrorist attack" as `perpetrator` (*"terrorist"*) and `action` (*"attack"*), although there were a few instances of both words together in the `action` slot. The tester always extracted both words together as `action` (see 0474, 0588, 0640).
- The tester extracted time periods into the `time` slot (*"[...] in less than 1 month, 11 leaders of the association of university students have been kidnapped by unidentified men [...]" (457), "[...] in the past 8 days in El Salvador, there have been several attacks [...]"(636)*); the expected results did not.
- For attributes holding information about people (`victim_target`, `victim_others_animate`, `perpetrator`), more information than expected was extracted. The tester's `victim_target` extraction for 437 (*"[...] 11 persons, including 2 newsmen, were wounded [...]"*) included "*were wounded*"; the expected result did not. In document 565 the expectation was *"children"*, the tester included additional information (*"her children, one of whom was wounded"*).
- The tester extracted only the person's full name (if provided). The expected results included titles and other information about the person if it was given before the name (e.g. *"Msgr. Oscar Arnulfo Romero" (038), "Major Horacio Fernandez Cutiello" (048), "social*

democratic leader Hector Oqueli Colindres" (038)).

Other notable differences between the human tester's extractions and the expected results:

- Some reports allowed different interpretations about who or what was the target and who/what was an other victim or the location of an attack. For example *"assassins paid by the cocaine cartels killed five Colombian personalities,including presidential hopeful Luis Carlos Galan"* (555): The tester extracted "*five Colombian personalities, including presidential hopeful Luis Carlos Galan*" as the `victim_target`. The `victim_target` in the example included only "*five Colombian personalities*" and "*Luis Carlos Galan*" was extracted as `victim_others_animate`.
  In the sentence stating *"the ARCE battalion indiscriminately bombed areas near Perkin"* (475), the tester put "*Perkin*" in the `town` slot while the example lists "*areas near Perkin*" as the `victim_target`.
- The `weapon` and `action` attributes lead to similar differences: In the description "*machine gun bursts*" (214) The tester put *"machine gun"* into the `weapon` slot. "*dynamite attack*" (622) was considered to be one `action`. The expectation split these words into two extractions, `weapon` and `act`. "*Four powerful dynamite explosions*" (568) are in the `weapon` slot of the expected results and in the `action` slot in the tester's.

The tester often extracted more terroristic acts from the documents than the number that was expected (most notably: 12 vs 7 for 638, 5 vs 2 for 639, 5 vs 0 for 295, 6 vs 2 for 142). In only a few documents did the expected results contain more extractions than the tester and there were never more than 2 additional acts expected. The number of documents with more than 3 extracted terroristic acts was 25 in the tester's extractions and 14 in the expectations.

4.2 Machine results

Quantitative Analysis: Total Results

ELIE's L2 learner achieved an average of about one fourth for both recall and precision with the regular runs. These overall results were worse than the human performance, which reached a recall of a little less than one half and precision of roughly one third.

The L1 learner, which produces good precision but low recall, scored an average recall of only 12% but its average precision was almost 40%, a little higher than that of the human extractions.

The following tables list ELIE's performance, broken down into L1, L2 learners, for the complete corpus and the pre-classified corpus. The values are based on the extractions for the attributes without the location attribute (i.e. `time`, `attackdate`, `perpetrator`, `perpetrator_number`, `perpetrator_organisation`, `action`, `weapon`, `victim_target`, `victim_others_animate`, `victim_others_inanimate`, `town`, `department`, `country`). The results achieved for the `location` attribute are covered in the analysis of the individual attributes.

The total number of true positives, false positives, and false negatives produced in one shuffle are the basis for the calculations of recall and precision ("microaverage") for the shuffle. The F-measure is the harmonic mean of recall and precision (i.e. F_1).

Table 8) Summary of performance for ELIE L1 learner, MUC-3 corpus (10 shuffles)

	***expected*[1]**	***made*[2]**	**Recall**	**Precision**	**F_1-Measure**
Smallest Value	*1539*	*477*	*11.13%*	*36.38%*	*17.06%*
Biggest Value	*1746*	*603*	*14.13%*	*43.61%*	*20.77%*
Average	***1672***	***518***	***12.31%***	***39.77%***	***18.78%***

[1]number of expected extractions [2]number of ELIE's extractions

Table 9) Summary of performance for ELIE L2 learner, MUC-3 corpus (10 shuffles)

	***expected*[1]**	***made*[2]**	**Recall**	**Precision**	**F_1-Measure**
Smallest Value	*1539*	*1390*	*22.31%*	*24.20%*	*23.21%*
Biggest Value	*1746*	*1641*	*25.93%*	*29.35%*	*27.12%*
Average	***1672***	***1486***	***24.01%***	***27.05%***	***25.42%***

[1]number of expected extractions [2]number of ELIE's extractions

The ranges of recall and precision values achieved in the different shuffles were small enough to suggest that other shuffles of the corpus would produce similar results. Therefore we considered these numbers to be a representations of ELIE's capabilities on the MUC-3 corpus.

ELIE made less extractions than the expected results contained. The L1 algorithm focuses on high precision but achieves only low recall. Therefore the number of made extractions was very low (less than one third of the number of expected extractions). The L1 learner achieved higher precision than the human performance (39.8% vs 36.5% respectively). Even the shuffle that produced the lowest precision was as good as the human tester's extractions.

The L2 learner was designed to improve recall of the L1 results but it reduces precision in the process. That is why the L2 algorithm made many more extractions than L1. As expected, L2 produced much better recall but also a significant decrease in precision, yet still resulted in a higher F-measure.

When considering only documents that contain at least one terroristic act, the percentage of extracted tokens in the corpus is bigger than that of the complete corpus.

The pre-classified corpus resulted in better performance; recall and precision were close to one third for the L2 learner. With 46%, the L1 learner's precision exceeded the human performance,

but the 17% recall was much worse.

Table 10) Summary of performance for ELIE L1 learner, pre-classified MUC-3 corpus (10 shuffles)

	expected[1]	made[2]	Recall	Precision	F_1-Measure
Smallest Value	*1595*	*560*	*14.82%*	*42.28%*	*22.09%*
Biggest Value	*1728*	*667*	*18.19%*	*49.21%*	*26.51%*
Average[3]	***1650***	***609***	***16.85%***	***45.71%***	***24.62%***

[1]number of expected extractions [2]number of ELIE's extractions

[3]The average is the average of the results of all 10 shuffles

Table 11) Summary of performance for ELIE L2 learner, pre-classified MUC-3 corpus (10 shuffles)

	expected[1]	made[2]	Recall	Precision	F_1-Measure
Smallest Value	*1595*	*1456*	*28.23%*	*30.41%*	*29.74%*
Biggest Value	*1728*	*1602*	*32.30%*	*34.57%*	*32.80%*
Average[3]	***1650***	***1528***	***30.19%***	***32.60%***	***31.34%***

[1]number of expected extractions [2]number of ELIE's extractions

[3]The average is the average of the results of all 10 shuffles

Apart from an overall better performance, the results of the pre-classified runs had the same characteristics as those of the regular runs concerning variation between shuffles, extraction number and differences between L1 and L2.

Individual Attributes

ELIE produced fewer extractions than the expected results, as the total number of extractions already showed. While the number of made extractions almost equaled that of the expectation for some attributes (e.g. time: 33 made, 35 expected; pre-classified: 33 vs 30), there was less than half as many made extractions as expected for others (perpetrator_number: 19 made vs 46 expected; pre-classified: 18 vs 44).

Table 12) Number of extractions made and expected, by attributes

	Average number of L2 extractions			
	Complete corpus		***pre-classified***	
Attribute	***expected[1]***	***made[1]***	***expected[1]***	***made[1]***
action	425	306	367	301
attackdate	219	174	178	163
country	124	60	117	79
department	71	43	60	41
location	270	141	282	169
perpetrator	245	151	233	156
perpetrator_number	46	19	44	18
perpetrator_organisation	143	84	122	75
time	35	33	30	33
town	219	143	212	167
victim_others_animate	130	73	129	72
victim_others_inanimate	50	10	52	12

	Average number of L2 extractions			
	Complete corpus		pre-classified	
Attribute	expected[1]	made[1]	expected[1]	made[1]
victim_target	500	289	499	316
weapon	148	103	139	95

[1]the average results are based on the sum of extractions of all shuffles. That is why the sums of these values do not equal the average number of extractions given above

There were significant differences in the performances for the different attributes, as you can see in tables 13 and 15 below. This is not surprising because the attributes have varied difficulties (e.g. time is limited to numbers, time units, and times of day only, while victim_target can be any person or object).

All attributes have in common that the L1 learner produced better precision than recall and the L2 improved recall but reduced precision. The improvement of recall is noticeable for all attributes – rather small for some attributes but it doubled for more than half of the attributes, and is even three times as big for four of the attributes (perpetrator_organisation, victim_others_animate, victim_others_inanimate, location). L2 delivered strongly decreased precision than L1 for some attributes (e.g. victim_target: from 44.07% down to 20.24%) while there was almost no change for others. Perpetrator_organisation and victim_others_inanimate both show a very big increase in recall with only a slight reduction of precision.

Since the L1 learner focused on precision at the cost of good recall, it produced clearly higher precision than recall for all attributes in our experiment. The difference between precision and recall was smaller for the L2 results because the L2 algorithm improved recall but lost some of L1's precision. Precision was only slightly bigger for town, perpetrator and action (precision was more than twice as good as recall for L1), and even worse than recall in the attackdate attribute.

Although the total results (all attributes combined) for the shuffles showed no big differences between best and worst performance (see tables 10 and 11), the highest and lowest recall/precision achieved by different runs differed significantly for each attribute (e.g. biggest difference: L1 perpetrator_number: one run with 0 precision, one with 60%).

The time attribute returned the best F-measure (average L2: 49.77%). The time information of the reported terroristic acts was usually limited to words that describe times of day, numbers, and time units. Most mentionings of times in the texts were related to the reported acts and were extracted as such in the training examples and expected results. Only few times of other events were reported. That is why it was a simple attribute to learn and extract, even though there were not many training examples (72 extractions from 650 documents). We had assumed attackdate to be similar to the time attribute because there were also only few ways to describe a date. However, the scores for this attribute were considerably worse than those for time (F-measure of 31.04 for L2).

The set of expressions used to describe action and weapon in the documents was rather limited as well, with some terms (e.g. attack, bomb) recurring very often. That is why ELIE achieved good results (F-measures of one third) for these two attributes.

Although the number of different words used for perpetrator_number data was also very limited, and there were many perpetrator_organisation extractions that recurred often, the scores for these attributes were low (L2 F-measure: 17.64%, 16.04% respectively).

With an F-measure of only 7.63%, ELIE produced the worst results for victim_others_inanimate. An explanation for this is that the extractions for this attribute

contained a large number of different words, there were many different ways to express victim information, and the differentiation of the extractions for the victim slots (`victim_target`, `victim_other_animate`, `victim_other_inanimate`) was not always unambiguous.

Table 13) Summary of performance for ELIE L1 & L2, ***MUC-3 corpus*** (10 shuffles)

	L1 learner			L2 learner		
	Recall[2]	**Precision[2]**	**F_1-Measure[2]**	**Recall[2]**	**Precision[2]**	**F_1-Measure[2]**
action						
smallest[1]	16.08%	34.78%	22.54%	28.04%	30.12%	29.04%
biggest[1]	25.00%	51.08%	32.19%	36.88%	38.20%	36.44%
average[1]	***20.92%***	***43.97%***	***28.30%***	***31.98%***	***34.44%***	***33.10%***
attackdate						
smallest[1]	9.09%	28.21%	13.75%	26.57%	22.89%	24.60%
biggest[1]	19.86%	48.28%	28.14%	45.00%	31.72%	35.39%
average[1]	***13.39%***	***37.25%***	***19.52%***	***34.11%***	***27.97%***	***30.63%***
perpetrator						
smallest[1]	9.14%	35.42%	14.88%	18.07%	22.73%	20.13%
biggest[1]	16.20%	56.10%	23.48%	27.93%	32.23%	28.09%
average[1]	***11.96%***	***43.04%***	***18.60%***	***22.90%***	***26.61%***	***24.52%***
perpetrator_number						
smallest[1]	0	0	0	2.78%	7.14%	4.00%
biggest[1]	12.50%	60.00%	18.18%	21.05%	61.54%	31.37%
average[1]	***6.61%***	***27.70%***	***10.14%***	***11.45%***	***23.84%***	***14.65%***
perpetrator_organisation						
smallest[1]	0	0	0	7.69%	11.89%	9.52%
biggest[1]	7.32%	33.33%	12.00%	22.09%	19.48%	19.79%
average[1]	***3.45%***	***16.28%***	***5.63%***	***15.86%***	***16.22%***	***15.76%***
time						
smallest[1]	23.53%	45.00%	32.14%	36.36%	41.03%	39.34%
biggest[1]	50.00%	72.22%	55.17%	60.61%	65.22%	58.82%
average[1]	***32.78%***	***58.58%***	***41.59%***	***47.46%***	***52.08%***	***49.01%***
victim_others_animate						
smallest[1]	1.12%	10.00%	2.15%	9.88%	13.89%	12.70%
biggest[1]	10.64%	52.63%	17.70%	23.40%	33.85%	27.67%
average[1]	***5.21%***	***27.10%***	***8.62%***	***17.20%***	***21.50%***	***18.86%***

	L1 learner			L2 learner		
	Recall[2]	Precision[2]	F_1-Measure[2]	Recall[2]	Precision[2]	F_1-Measure[2]
victim_others_inanimate						
smallest[1]	0	0	0	0	0	0
biggest[1]	2.50%	50.00%	4.65%	10.00%	22.22%	13.33%
average[1]	***0.72%***	***13.33%***	***1.35%***	***3.05%***	***12.21%***	***4.74%***
victim_target						
smallest[1]	4.97%	36.00%	8.74%	15.85%	15.99%	16.61%
biggest[1]	10.23%	54.17%	16.79%	21.02%	24.91%	22.80%
average[1]	***7.44%***	***44.07%***	***12.70%***	***17.74%***	***20.24%***	***18.86%***
weapon						
smallest[1]	12.82%	34.09%	18.63%	27.35%	35.64%	31.58%
biggest[1]	23.39%	50.00%	31.69%	37.62%	42.27%	39.81%
average[1]	***19.44%***	***45.31%***	***27.15%***	***31.63%***	***37.92%***	***34.42%***
country						
smallest[1]	1.10%	6.67%	1.98%	3.30%	6.45%	4.44%
biggest[1]	11.11%	42.11%	17.58%	18.06%	20.00%	17.57%
average[1]	***4.50%***	***21.07%***	***7.34%***	***9.96%***	***13.14%***	***11.14%***
department						
smallest[1]	5.36%	23.08%	8.70%	12.90%	26.67%	17.39%
biggest[1]	28.26%	47.06%	35.14%	36.96%	39.47%	36.96%
average[1]	***13.38%***	***37.92%***	***19.12%***	***24.93%***	***31.97%***	***27.68%***
town						
smallest[1]	5.56%	22.22%	8.89%	15.89%	18.61%	17.14%
biggest[1]	16.35%	41.38%	22.64%	31.45%	31.66%	30.77%
average[1]	***10.58%***	***31.54%***	***15.75%***	***25.09%***	***26.95%***	***25.92%***
location						
smallest[1]	1.16%	12.50%	2.13%	7.56%	9.15%	8.41%
biggest[1]	7.33%	40.74%	12.43%	14.38%	20.16%	16.77%
average[1]	***3.38%***	***25.38%***	***5.92%***	***12.26%***	***15.12%***	***13.47%***

[1]The values for smallest (biggest) recall (precision) are the minimum (maximum) of the recall (precision) scores of the 10 shuffles. Smallest (biggest) recall, precision,F_1 were not necessarily achieved in the same shuffle.
[2] average recall (precision, F-measure) = sum of recall (precision, F-measure) of the shuffles divided by the number of shuffles

As expected, ELIE's results demonstrated that the specific geographic attributes combined yielded better performance than the general attribute `location`. Recall and precision of the `town`

and `department` extractions were better than those for `location`. The extractions for these attributes contained names of towns/departments, which usually have easily identifiable starts and ends (resulting in only few partial matches, see partial match analysis on page 53).

The `location` slot fills, on the other hand, could contain the names of towns, departments, countries, streets, and other places as well as nouns such as "house", and often held text passages that combined more than one of these location descriptions (e.g. "the victim's house in San Salvador"). In location extractions that consisted of more than one name, the starts and ends of an extraction could be ambiguous.

Just like `town` and `department`, the `country` attribute asks for geographic names. As the extraction task was to extract terroristic acts in South America, there were only a few names that qualified for the `country` slot. This would suggest that ELIE (and any other IE system) would perform well on the `country` attribute, just like it did on `town` and `department`.

However, the scores for `country` were worse than those for `location` (L2 F-measure: 11.14% for country vs 13.47% for location). This phenomenon was found in the human results as well, and one of the explanations for the bad performance there applied here, too: The training examples contained inconsistencies and errors, making it difficult to construct a data model that mirrored the patterns used for extracting.

Table 14) Summary of performance on place attributes; L1 & L2, MUC-3 corpus

	Recall	Precision	F_1-Measure		Recall	Precision	F_1-Measure
country, department, town combined L1				***country, department, town combined L2***			
	13.05%	*40.51%*	*19.74%*		*22.40%*	*28.04%*	*24.90%*
location L1				***location L2***			
	3.38%	25.38%	5.92%		12.26%	15.12%	13.47%

As the total results for the entire corpus already showed, ELIE's performance on the pre-classified corpus was better than that on the whole corpus.

Table 15) Summary of performance for ELIE L1 & L2, ***pre-classified MUC-3 corpus*** (10 shuffles)

Attribute	L1 learner			Attributes	L2 learner		
	Recall	Precision	F_1-Measure		Recall	Precision	F_1-Measure
action				**action**			
smallest[1]	22.58%	45.16%	30.11%	smallest[1]	36.78%	38.80%	38.10%
biggest[1]	30.49%	55.00%	39.14%	biggest[1]	44.21%	45.74%	44.96%
average[2]	***26.44%***	***50.29%***	***34.60%***	***average[2]***	***39.83%***	***42.53%***	***41.11%***
attackdate				**attackdate**			
smallest[1]	19.05%	41.89%	26.79%	smallest[1]	40.69%	34.09%	37.34%
biggest[1]	29.85%	59.09%	39.22%	biggest[1]	52.24%	44.12%	46.90%
average[2]	***23.46%***	***51.50%***	***32.17%***	***average[2]***	***44.78%***	***38.97%***	***41.58%***

Attribute	L1 learner			Attributes	L2 learner		
	Recall	Precision	F_1-Measure		Recall	Precision	F_1-Measure
perpetrator				**perpetrator**			
smallest[1]	11.45%	35.19%	17.27%	smallest[1]	24.10%	27.03%	25.48%
biggest[1]	20.00%	64.71%	29.07%	biggest[1]	32.79%	35.29%	33.99%
average[2]	***16.77%***	***48.32%***	***24.76%***	***average[2]***	***27.97%***	***31.14%***	***29.38%***
perpetrator_number				**perpetrator_number**			
smallest[1]	0	0	0	smallest[1]	4.55%	16.67%	7.14%
biggest[1]	21.88%	60.00%	29.79%	biggest[1]	28.13%	40.00%	30.51%
average[2]	***7.73%***	***28.84%***	***11.78%***	***average[2]***	***14.46%***	***27.24%***	***18.35%***
perpetrator_organisation				**perpetrator_organisation**			
smallest[1]	4.40%	19.05%	7.14%	smallest[1]	11.91%	17.54%	14.18%
biggest[1]	13.75%	45.00%	20.37%	biggest[1]	32.10%	32.10%	32.10%
average[2]	***9.57%***	***34.56%***	***14.81%***	***average[2]***	***21.75%***	***24.09%***	***22.75%***
time				**time**			
smallest[1]	20.00%	50.00%	28.57%	smallest[1]	25.71%	37.50%	30.51%
biggest[1]	56.76%	100%	65.63%	biggest[1]	70.27%	75.00%	68.42%
average[2]	***40.95%***	***73.47%***	***50.91%***	***average[2]***	***54.60%***	***60.20%***	***56.31%***
victim_others_animate				**victim_others_animate**			
smallest[1]	1.18%	7.69%	2.04%	smallest[1]	1.49%	12.20%	11.83%
biggest[1]	13.25%	58.33%	20.37%	biggest[1]	27.91%	31.17%	29.45%
average[2]	***6.35%***	***28.56%***	***10.22%***	***average[2]***	***17.83%***	***22.02%***	***19.53%***
victim_others_inanimate				**victim_others_inanimate**			
smallest[1]	0	0	0	smallest[1]	0	0	0
biggest[1]	2.33%	33.33%	4.35%	biggest[1]	6.98%	33.33%	10.35%
average[2]	***0.23%***	***3.33%***	***0.43%***	***average[2]***	***3.29%***	***12.76%***	***5.10%***
victim_target				**victim_target**			
smallest[1]	7.95%	34.67%	12.94%	smallest[1]	18.71%	19.33%	19.02%
biggest[1]	12.31%	55.17%	19.32%	biggest[1]	24.53%	25.90%	25.20%
average[2]	***10.00%***	***45.53%***	***16.34%***	***average[2]***	***21.70%***	***22.22%***	***21.92%***
weapon				**weapon**			

Attribute	L1 learner			Attributes	L2 learner		
	Recall	Precision	F_1-Measure		Recall	Precision	F_1-Measure
smallest[1]	15.00%	40.00%	21.82%	smallest[1]	27.50%	34.02%	30.41%
biggest[1]	23.44%	62.79%	33.54%	biggest[1]	36.00%	50.00%	41.86%
average[2]	***20.35%***	***50.06%***	***28.87%***	***average[2]***	***32.02%***	***42.48%***	***36.42%***
country				**country**			
smallest[1]	4.88%	13.33%	7.55%	smallest[1]	15.07%	18.03%	16.42%
biggest[1]	15.94%	33.33%	20.95%	biggest[1]	33.33%	31.25%	28.17%
average[2]	***8.74%***	***21.90%***	***12.22%***	***average[2]***	***24.25%***	***22.91%***	***23.31%***
department				**department**			
smallest[1]	10.42%	38.46%	16.39%	smallest[1]	23.53%	35.29%	28.23%
biggest[1]	27.45%	61.11%	37.33%	biggest[1]	38.98%	50.00%	40.43%
average[2]	***19.36%***	***51.28%***	***27.88%***	***average[2]***	***32.08%***	***41.76%***	***36.05%***
town				**town**			
smallest[1]	12.42%	29.69%	17.84%	smallest[1]	28.11%	27.22%	28.11%
biggest[1]	22.70%	48.49%	30.92%	biggest[1]	45.39%	40.51%	42.81%
average[2]	***16.85%***	***37.77%***	***23.28%***	***average[2]***	***35.29%***	***32.28%***	***33.68%***
location				**location**			
smallest[1]	1.71%	11.54%	2.98%	smallest[1]	13.71%	12.57%	13.17%
biggest[1]	6.94%	34.62%	11.43%	biggest[1]	22.29%	22.37%	21.05%
average[2]	***4.52%***	***24.06%***	***7.57%***	***average[2]***	***16.39%***	***16.19%***	***16.25%***

[1]The values for smallest (biggest) recall (precision) are the minimum (maximum) of the recall (precision) scores of the 10 shuffles. Smallest (biggest) recall, precision,F_1 were not necessarily achieved the same shuffle.
[2] average recall (precision, F-measure) = sum of recall (precision, F-measure) of the shuffles divided by the number of shuffles

The L1 algorithm returned precision and recall of 0 in nine of the ten runs for the `victim_others_inanimate` attribute on the pre-classified corpus, while it achieved a precision of 50% in two shuffles of the whole corpus and 33% in another. Recall was a little over 2% in all three runs. The other seven runs also resulted in 0 recall and precision.

With the exception of `victim_other_inanimate` L1 and `country` L1, the pre-classification improved performance for all attributes, more for some than for others. There was no correlation between the improvement of precision/recall and number of extractions for the attribute in the training examples. For the L1 learner, the relative advancement of recall tended be higher than that of precision for the pre-classified corpus. For the L2 learner, there was no such tendency.

Number of Tokens Per Extraction

The analysis of the number of tokens per extraction for the machine performance were based on the extractions for one shuffle; the one that was also used for the qualitative analysis. As there was no unusual deviation between the results of this shuffle and the results achieved in the other

shuffles, these results could be considered to be a representation of ELIE's performance.

Table 16) Number of tokens per extraction by attribute

Number of tokens	Attribute	Number of extractions	Expected results[2]	Difference
1	total[1]	562	776	-214
2	total[1]	394	464	-70
3	total[1]	136	155	-19
4	total[1]	60	72	-12
5	total[1]	64	45	19
6	total[1]	23	22	1
7	total[1]	20	20	0
8	total[1]	13	14	-1
9	total[1]	8	6	2
10	total[1]	9	3	6
11	total[1]	10	5	5
12	total[1]	7	1	6
13	total[1]	0	1	-1
15	total[1]	2	1	1
16	total[1]	5	3	2
17	total[1]	1	2	-1
19	total[1]	7	1	6
20	total[1]	0	1	-1
1	action	217	260	-43
2	action	27	33	-6
3	action	12	9	3
4	action	5	2	3
5	action	4	2	2
10	action	0	1	-1
1	attackdate	18	27	-9
2	attackdate	125	93	32
3	attackdate	21	18	3
4	attackdate	1	1	0
6	attackdate	0	1	-1
7	attackdate	0	1	-1
1	country	26	66	-40
2	country	9	15	-6
1	department	40	38	2
2	department	5	6	-1
3	department	1	2	-1
1	location	9	23	-14
2	location	38	39	-1
3	location	20	19	1
4	location	14	16	-2
5	location	13	15	-2
6	location	13	14	-1
7	location	4	6	-2
8	location	8	5	3
9	location	4	7	-3
10	location	5	2	3
11	location	0	2	-2

Number of tokens	Attribute	Number of extractions	Expected results[2]	Difference
12	location	3	1	2
13	location	1	1	0
14	location	3	1	2
15	location	1	2	-1
16	location	2	2	0
1	perpetrator	45	84	-39
2	perpetrator	54	54	0
3	perpetrator	17	16	1
4	perpetrator	3	3	0
5	perpetrator	0	2	-2
6	perpetrator	1	1	0
7	perpetrator	3	1	2
8	perpetrator	0	2	-2
9	perpetrator	0	1	-1
16	perpetrator	0	1	-1
17	perpetrator	0	1	-1
1	perpetrator_number	12	36	-24
2	perpetrator_number	1	2	-1
1	perpetrator_organisation	8	39	-31
2	perpetrator_organisation	9	14	-5
3	perpetrator_organisation	11	8	3
4	perpetrator_organisation	9	7	2
5	perpetrator_organisation	35	13	22
6	perpetrator_organisation	1	1	0
7	perpetrator_organisation	0	1	-1
8	perpetrator_organisation	0	2	-2
9	perpetrator_organisation	0	1	-1
1	time	45	35	10
2	time	0	4	-4
3	time	0	1	-1
4	time	0	3	-3
6	time	0	1	-1
1	town	85	85	0
2	town	43	49	-6
3	town	2	4	-2
4	town	0	4	-4
1	victim_others_animate	2	5	-3
2	victim_others_animate	25	50	-25
3	victim_others_animate	2	15	-13
4	victim_others_animate	6	4	2
5	victim_others_animate	3	6	-3
6	victim_others_animate	0	1	-1
7	victim_others_animate	2	4	-2
8	victim_others_animate	1	1	0
9	victim_others_animate	0	1	-1
11	victim_others_animate	1	1	0
16	victim_others_animate	0	1	-1
1	victim_others_inanimate	2	9	-7
2	victim_others_inanimate	11	19	-8

Number of tokens	***Attribute***	***Number of extractions***	***Expected results[2]***	***Difference***
3	victim_others_inanimate	2	3	-1
4	victim_others_inanimate	0	4	-4
5	victim_others_inanimate	0	2	-2
8	victim_others_inanimate	0	2	-2
20	victim_others_inanimate	0	1	-1
1	victim_target	10	39	-29
2	victim_target	60	82	-22
3	victim_target	65	70	-5
4	victim_target	29	37	-8
5	victim_target	21	18	3
6	victim_target	21	17	4
7	victim_target	14	11	3
8	victim_target	12	7	5
9	victim_target	8	3	5
10	victim_target	9	2	7
11	victim_target	9	4	5
12	victim_target	7	1	6
15	victim_target	2	1	1
16	victim_target	5	1	4
17	victim_target	1	1	0
19	victim_target	7	1	6
1	weapon	52	53	-1
2	weapon	25	43	-18
3	weapon	3	9	-6
4	weapon	7	7	0
5	weapon	1	2	-1
7	weapon	1	2	-1
13	weapon	0	1	-1

[1] the total results refer to the extractions for all attributes *excluding* the location attribute

[2] the frequency of token numbers for the expected results refers to the same documents that the extractions were made for; they are not an average based on the number of extractions for the entire corpus from the previous chapter

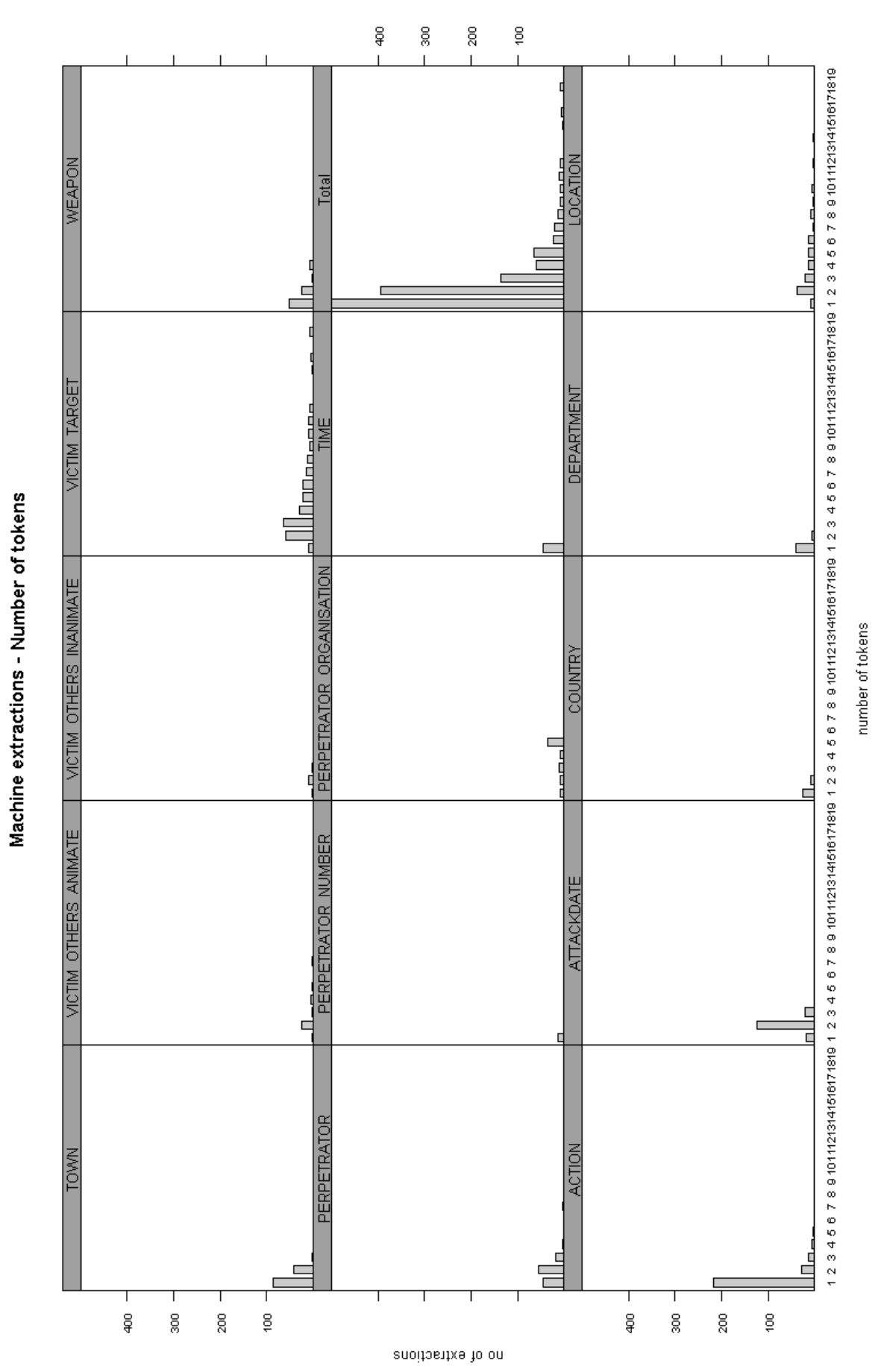
As ELIE's algorithm matched its predicted start and end tokens based on the number of tokens found in the training example extractions, the distribution of number of tokens per extraction for ELIE's results is overall similar to that of the expectations. However, ELIE made considerably fewer extractions containing less than five tokens than the training examples contained.

The fact that both ELIE's results and the expectations included only few extractions made up of many tokens, no generalization can be made about the performance for those.

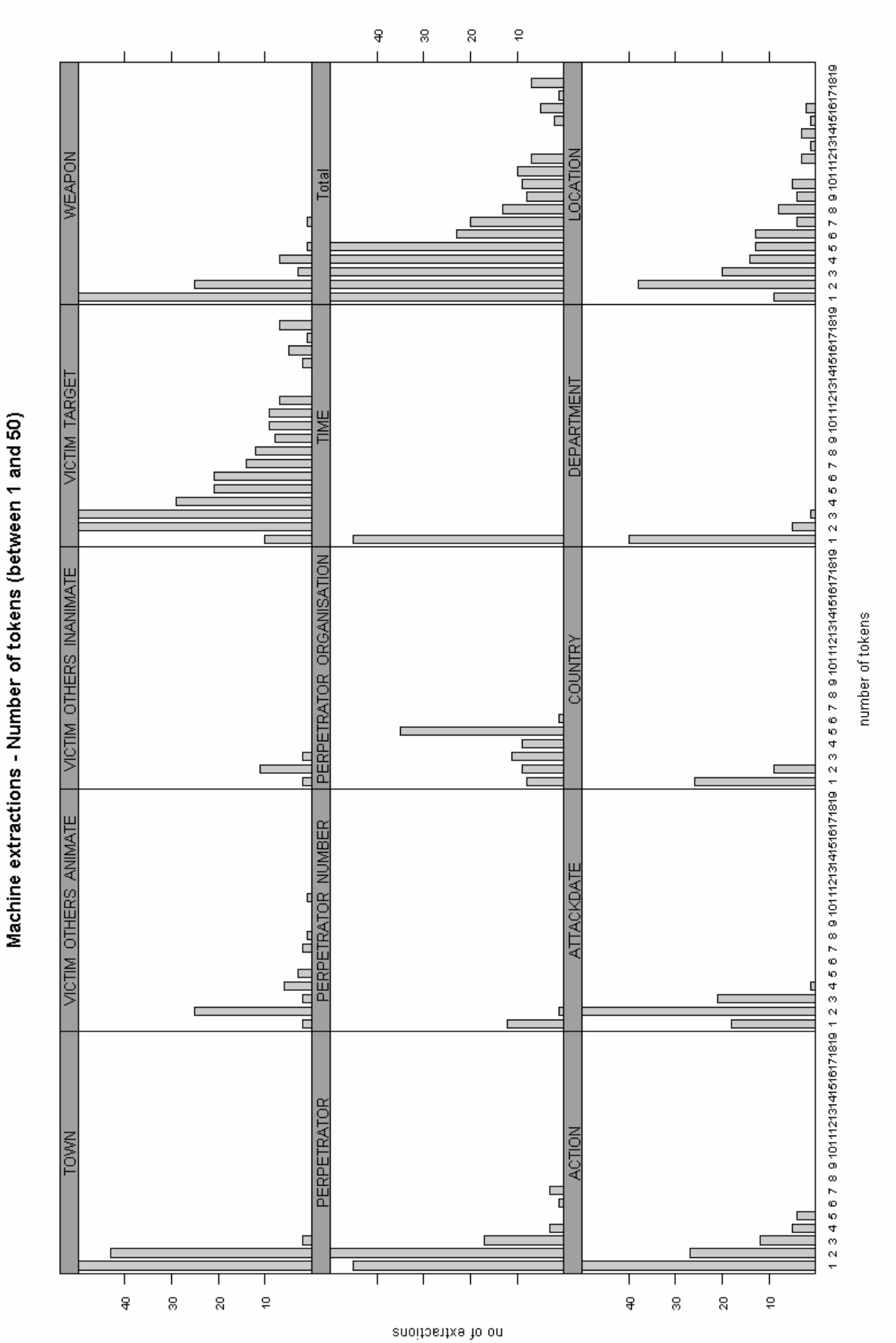
One noticeable difference in the plots for the distribution of token numbers in expected results (plots 1 and2) and ELIE's (plots 8 and 9) was that ELIE produced more extractions with many tokens than expected and less extractions with only few tokens for `victim_target`.

Please note that the plot for the expected results was based on the entire corpus (650 documents) and the machine performance evaluated the extractions of only 325 documents. Due to the lower number of extractions of ELIE's results the y-axes of the plots have different maxima than those for the expectation plots.

Plot 8) Machine performance: Distribution of extractions per number of tokens (by attributes) overview

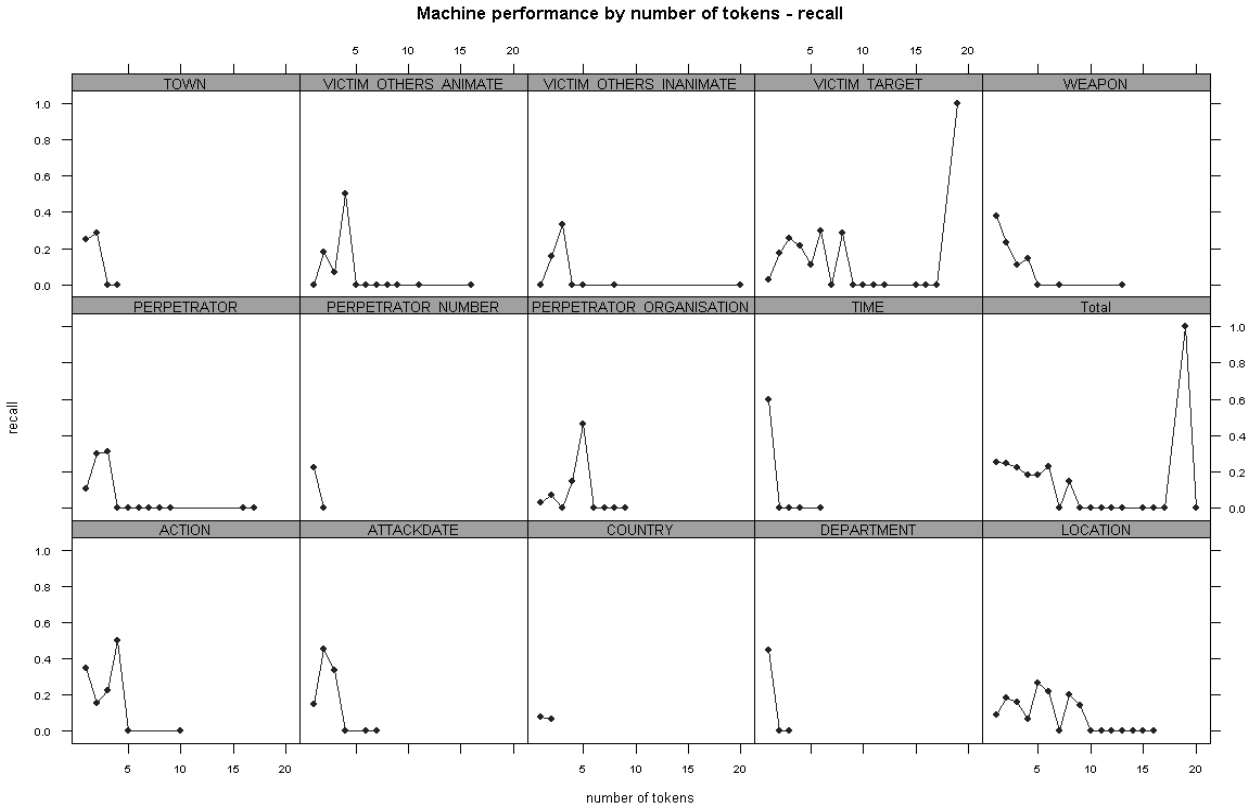


Plot 9) Expected Results: Distribution of extractions per number of tokens (by attributes)
Detailed look at extraction frequencies between 1 and 50:

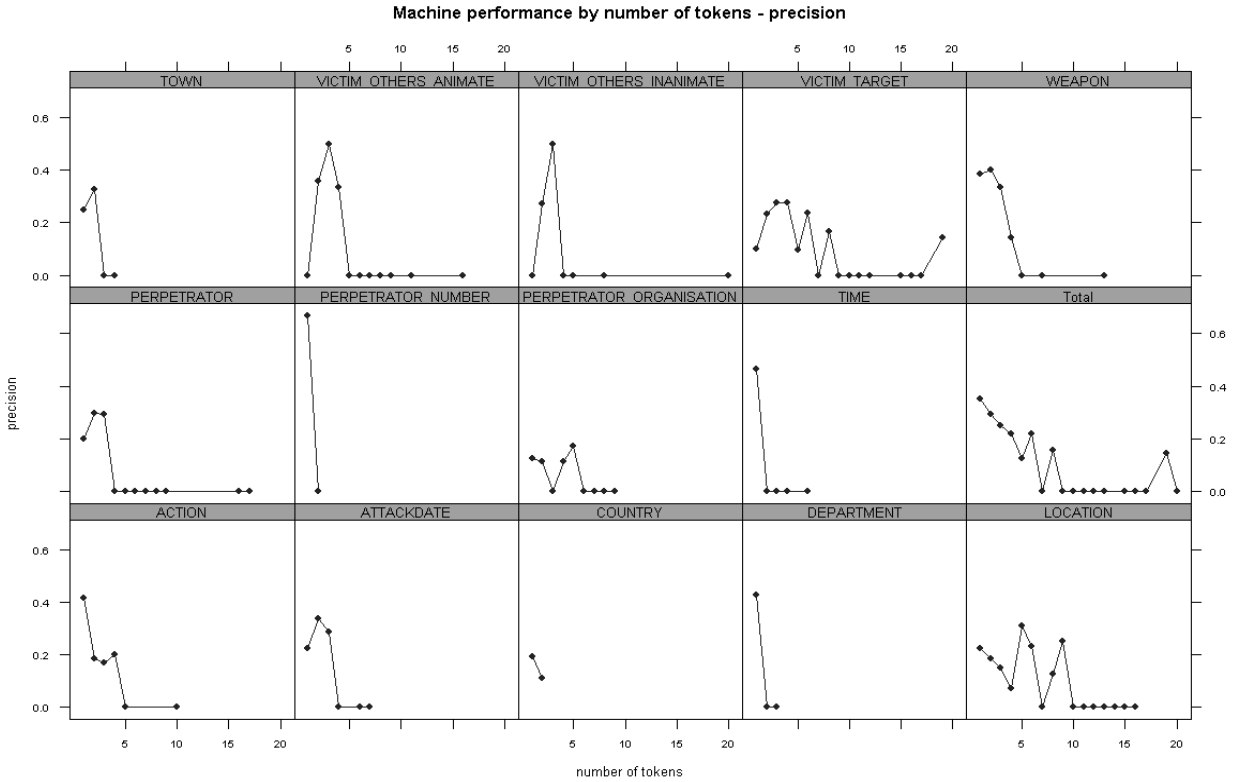


The performance measures broken down into number of tokens show that ELIE's performance was better for extractions made up of few tokens – but not always best for one-token-extractions. However, the low number of extractions with many tokens did not allow to a conclusion about ELIE's performance for long extractions.

Plot 10) Machine performance: Recall by number of tokens (by attribute)



Plot 11) Machine performance: Precision by number of tokens (by attribute)



Partial Matches

We differentiated between the same types of partial matches that we used to classify the partials of the human performance:

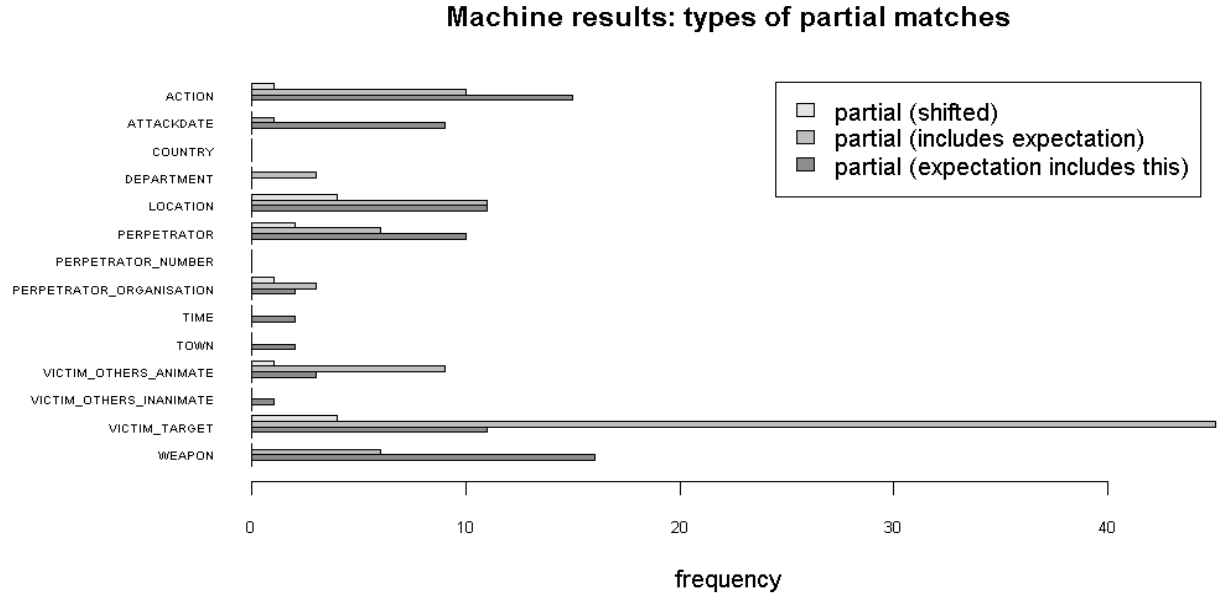
- made extraction is included in the expectation (i.e. later/same start and earlier/same end but not a match) – put simply: "less extracted"
- made extraction includes the expectation (i.e. earlier/same start and later/same end but not a match) – "more extracted"
- made extraction is shifted compared to the expectation (i.e. starts and ends earlier or starts and ends later; both contain at least one token that the other does not)

Table 17) Machine performance: Partial matches, complete corpus

Attribute	*Included in expectation*	*Includes expectation*	*Shifted*	*Percentage of extractions*
action	15	10	1	9.7%
attackdate	9	1	0	6.0%
country	0	0	0	0
department	0	3	0	6.5%
location	11	11	4	24.8%
perpetrator	10	6	2	13.6%
perpetrator_number	0	0	0	0
perpetrator_organisation	2	3	1	8.2%
time	2	0	0	4.4%
town	2	0	0	1.5%
victim_others_animate	3	9	1	27.7%
victim_others_inanimate	1	0	0	5.0%
victim_target	11	45	4	18.8%
weapon	16	6	0	23.4%
total	82	94	13	12.7%

number of partials with same start	35
number of partials with same end	103

Plot 12) Machine performance: Partial matches, complete corpus



As ELIE predicted start and end tags independently and afterwards matched starts and ends to form an extraction, it was more likely that a partial match shared either start tag or end tag with

the expected extraction than sharing neither. That is why 138 of the 189 partials (over 70%) had one tag in common with the extraction they matched partially.

To the observation that ELIE made more long extractions for `victim_target` (see token number analysis above) we could add that many of the long extractions contain expected facts.

Operating Times

Each ELIE run received with one target structure attribute and a list of shuffles as input. The shuffles were processed consecutively and as independent tasks. This means that the operating time for ten shuffles is equal to the sum of the operating times for ten runs with one shuffle each. A run included training, L1 fact extraction, subsequent L2 fact extraction, and a printout of the results.

The ELIE runs for this experiment were carried out on computers with a Linux operating system. No other resource intensive processes were executed at the time the runs were started.

The run times below do not include document list and shuffle creation, and pre-classification.

Table 18) Machine performance: Operating times by attributes (all documents)

Complete corpus			
	Average number of extractions		***Average run time per shuffle in minutes***
Attribute	***in training documents***	***made***	
action	321	306	114
attackdate	143	174	66
country	74	60	68
department	53	43	63
location	170	141	60
perpetrator	173	151	87
perpetrator_number	37	19	53
perpetrator_organisation	89	84	78
time	37	33	60
town	154	143	108
victim_others_animate	87	73	75
victim_others_inanimate	41	10	66
victim_target	322	289	126
weapon	123	103	78
total	*1668*	*1629*	*1102 + 15 (pre-processing)*

Table 19) Machine performance: Operating times by attributes (pre-processed corpus)

Pre-classified corpus			
	Average number of extractions		***Average run time per shuffle in minutes***
Attribute	***in training documents***	***made***	
action	329	301	56

Pre-classified corpus

Attribute	*Average number of extractions*		*Average run time per shuffle in minutes*
	in training documents	*made*	
attackdate	144	163	35
country	79	79	32
department	55	41	30
location	174	169	30
perpetrator	175	156	44
perpetrator_number	36	18	24
perpetrator_organisation	91	75	26
time	36	33	23
town	155	167	56
victim_others_animate	87	72	35
victim_others_inanimate	42	12	24
victim_target	326	316	44
weapon	121	95	29
Total	*1850*	*1697*	*488 + 15*[2]

[1]The average results are based on the sum of extractions of all shuffles
[2] *There was no additional preprocessing for the pre-classified runs*

There was no relationship between the number of extractions (training+made) and the operating time (e.g. attackdate and department have equal run time but there were more than twice as many training and made extractions for attackdate than for department) .

Qualitative Analysis

We analyzed the results of one of the shuffles by viewing them in the new evaluation module of XTract (see chapter 6. Implementation: Evaluation Module for XTract).

A great number of ELIE's incorrect extractions had the correct part of speech for the respective attribute and often fitting values for simple attributes (words describing a violent act for `action`; names of an organization in the `perpetrator_organisation` slot; etc) even if the context was not always that of a terroristic act. There were also a number of slots filled with co-references to correct facts.

Common errors in ELIE's extractions

In the training examples and human extractions, whole terroristic acts were extracted because the database contained relations ("rows") which had one field for each attribute ("column") of the template. Attribute extractions belonging to the same act were extracted into one database relation. There were no acts with only one slot filled; there were a few acts with two attribute entries but the majority of acts contained more facts.

This also meant that in the human fact extraction task only one slot per attribute was available for a terroristic act. If a fact was given multiple times or more than one fact was given that could fit in one slot, only one of these could be extracted and the person extracting the information had to choose one of them.

However, the relations between attributes were lost when the database extractions of the training examples were exported into documents with attribute tags. These annotated texts were the input format for ELIE.

ELIE's algorithm did not extract terroristic acts but individual attributes. This approach allowed the extraction of more than one fact per attribute within one terroristic act and the extraction of only one fact of a terroristic act. These two extraction properties were present in ELIE's results of our experiment.

ELIE's results contained several individual extractions that were not related to other extractions in the same document. Some of them provided facts about terroristic acts and were correct or partial matches, while others were taken from paragraphs that did not contain information on any kind of terroristic or other violent act.

- An example for the former is the partially correct `attackdate` in "*Over 20 bomb explosions were heard in the predawn and early morning hours today in San Salvador*" (138).

  The `time` extraction in "*They will take Lan-Chile flight 141 that departs at approximately 2100 from Santiago and goes directly to New York.*" and an `action` in "*the main objective of the trip is to achieve the lifting of the ban on imported Chilean fruits. U.S. officials announced the ban [...] following the discovery that some Chilean grapes were laced cyanide*" were the only two extractions made in document 123, which did not contain any expected results.

Information extraction usually does not allow overlapping extractions, i.e. each token/character in a document can be included in no more than one extraction. The training examples and human extractions followed this rule[1].

ELIE, however, does not apply this concept. ELIE also allowed overlapping extractions for the same attribute. In addition to that it was possible that ELIE's extractions for different attributes overlapped, because training and fact extraction was done for only one attribute at a time and the extractions already made for the other attributes were not provided.

ELIE's results for the MUC-3 corpus contained several of these overlapping extractions.

1 Note: The location extractions could overlap with extractions for town/department/country because they were part of different target structures. The extractions for both target structures were treated as different sets of extractions.

- Overlapping extractions of different attributes could be found in the sentence "*One person was killed and three others injured tonight as the result of a bomb explosion in downtown San Salvador. the explosion took place on first avenue west, between the Vicas store and central reserve bank*." (097), where "*bomb*" was correctly identified as `weapon` and everything the rest of the sentence starting with "*bomb*" was extracted as `victim_target`. Both "*two bomb*" and "*bomb*" were extracted as `weapon` in "*two bomb attacks were carried out [...]*" (024). This is one of many examples of overlapping extractions for the same attribute.
- The extractions made by ELIE included expressions that contained parts of two paragraphs of a document, e.g. "*the second attack occurred at 2335 [0335 GMT on 12 January], just after the cabinet members had left government house where they had listened to the presidential message.* [end of paragraph] *A bomb was placed outside government house in the parking lot [...]*" (024; other tokens of this part of the text were included in ELIE's or the expected extractions but are not marked here). There were no extractions spanning over the end of a paragraph in the training examples.
- In 45 of its `victim_target` slots, ELIE included more information in the extractions than necessary (see above analysis of partial matches). This additional information often included facts of other attributes, e.g. the `perpetrator` extraction in the sentence "*[...] justice minister Julio Alfredo Samayoa said today all the evidence [...] demonstrates that Dr Antonio Regalado was the perpetrator of the murder of Msgr Romero.*" (062) includes `action` ("*murder*") and `victim_target` ("*Msgr Romero*"), which were also identified correctly.
- Fills for other attributes also contained more information than needed (`weapon`: "*[...]because there were reports that three other car bombs had not yet exploded*" (223)).
- While the training examples contained only extractions of whole words, ELIE's results included partial words that omitted the suffix, e.g. "*soldiers murdered 17 fishermen*" (367) .

There were a few cases of wrong part-of-speech or meaning assignments for homographs, resulting in extractions that a person would not be likely to make.

- ELIE extracted "*they may*" as an `attackdate` in the sentence " *[...] they are allowed to carry weapons [...] although they may not participate in offensive operations.*" (147).
- In the sentence "*Manuel Francisco Madero [...] was assassinated this evening [...]*" (631) "*evening*" was apparently considered to be an inflection of "to even (something)", resulting in an `action` extraction of "this evening" (it was also correctly identified as attackdate).
- "*little over one month*" was extracted as `location` in "*[...] escalation of violence in which over 70 bombs have exploded in a little over 1 month*" (523).

Single clearly erroneous training example extractions, such as "*22 March*" as a `location` (see Weaknesses of the Training Examples & Expected Results, page 23) might have been the basis for these extractions.

4.3 Comparison Human Performance – Machine Performance

Note: Although the machine analysis in the previous chapter examined the results of ELIE's L1 and L2 learners, we only look at the L2 results for the comparison because the L2 performance (as evidenced by the higher F-measure) was better than L1's.

Comparison of quantitative evaluation data

The human performance F-measure of 40% was significantly better than that of the machine results (25%). The program runs on the pre-classified corpus produced better scores than those on the whole corpus, but they still did not reach the same values as the human extractions.

Table 20) Human & machine performances, total results

	Recall	***Precision***	***F_1-Measure***
Human tester			
Total (with town,department, country)*	47.2 %	36.5 %	41.2 %
ELIE L2, complete corpus			
Average total	24.01%	27.05%	25.42%
ELIE L2, pre-classified corpus			
Average total	30.19%	32.60%	31.34%

While the human extractions received better overall scores than the machine's, ELIE scored higher measures on four of the attributes, as the following table shows.

Table 21) Human & machine performances, individual attributes

Human tester				ELIE, complete corpus L2 learner[1]				ELIE pre-classified corpus L2 learner[1]			Example extract-ions[2]
Recall	Preci-sion	F_1		Recall	Preci-sion	F_1		Recall	Preci-sion	F_1	
50.2%	41.0%	**45.1%**	action	31.98%	34.44%	33.10%	action	39.83%	42.53%	41.11%	650
65.2%	40.2%	**49.8%**	attackdate	34.11%	27.97%	30.63%	attackdate	44.78%	38.97%	41.58%	285
48.2%	37.8%	**42.3%**	perpetrator	22.90%	26.61%	24.52%	perpetrator	27.97%	31.14%	29.38%	348
36.8%	36.9%	**36.8%**	perpetrator_number	11.45%	23.84%	14.65%	perpetrator_number	14.46%	27.24%	18.35%	72
52.3%	32.7%	**40.2%**	perpetrator_organisation	15.86%	16.22%	15.76%	perpetrator_organisation	21.75%	24.09%	22.75%	175
65.9%	38.2%	48.3%	time	47.46%	52.08%	49.01%	time	54.60%	60.20%	**56.31%**	72
42.7%	28.4%	**34.1%**	victim_others_animate	17.20%	21.50%	18.86%	victim_others_animate	17.83%	22.02%	19.53%	175
32.5%	19.7%	**24.5%**	victim_others_inanimate	3.05%	12.21%	4.74%	victim_others_inanimate	3.29%	12.76%	5.10%	84
44.4%	33.3%	**38.1%**	victim_target	17.74%	20.24%	18.86%	victim_target	21.70%	22.22%	21.92%	649
59.8%	52.7%	**56.0%**	weapon	31.63%	37.92%	34.42%	weapon	32.02%	42.48%	36.42%	246
46.5%	37.3%	**41.4%**	town	25.09%	26.95%	25.92%	town	35.29%	32.28%	33.68%	308
32.6%	25.4%	28.6%	department	24.93%	31.97%	27.68%	department	32.08%	41.76%	**36.05%**	108
4.9%	16.0%	7.5%	country	9.96%	13.14%	11.14%	country	24.25%	22.91%	**23.31%**	153
16.8%	14.3%	15.4%	location	12.26%	15.12%	13.47%	location	16.39%	16.19%	**16.25%**	341

[1] averages of the results from all program run (different shuffles).
2 The number of extractions in the training examples/expected results (all 650 documents).

Bold numbers mark the highest F-measure for the attribute.

The fact that for each attribute there were more extractions made by the human tester than the expected result contained, suggested a tendency towards recall rather than precision (extraction of everything in every document leads to 100% recall). The extractions created by the human tester did in fact achieve higher recall than precision (exceptions: `perpetrator_number` and the problematic `country` attribute), while ELIE scored higher precision than recall on all attributes except `attackdate`, `town`, `country`, `location`. This is due to ELIE's focus on precision (although the L2 learner improves recall while losing some precision).

The performance measure of ELIE's extraction exceeded the human results for the attributes `time`, `department`, `country`, and `location`. The human tester's extractions reached higher F-measures on the other ten attributes. While ELIE's pre-classified runs reached F-measures of a little over 40% for `action` and `attackdate` and the human results were better by only a few percent – 45% and 50% respectively – the superiority of the person's F-measures for the other attributes was more significant. The performance for `action` and `attackdate` was better than that for most of the other attributes for both the machine and the human tester.

The machine performance F-measure on the pre-classified corpus for `perpetrator_number` (18%) and `victim_others_inanimate` (5%), attributes for which only few training examples existed, was poor compared to the human tester's (37% for the former and 25% for the latter). This could be explained by the fact that having more instances to learn from provided a better data model for machine predictions. Human readers, on the other hand, can use their knowledge about an attribute to identify relevant facts even if the training examples do not contain enough data to construct an extraction pattern. ELIE's F-measure for `victim_target` (22% for pre-classified) was also much lower than that of the human data (38%).

As opposed to the findings for `perpetrator_number` and `victim_others_inanimate`, there were many training examples to learn from for `victim_target`. Since this attribute covered a semantically and syntactically broad field of relevant facts (including expressions describing people, people's names, organizations, places; only one subject/object or enumerations of several) and facts that had the same parts of speech as the extractions for other attributes (e.g. names, organizations, objects), even a data model based on a large amount of data could not give a clear outline for which tokens could be classified as `victim_target` and which did not.

The observation we made for attributes with only few examples was not applicable to the results we received for the `time` attribute. There were only 72 `time` extractions in the training examples and expected results combined, meaning that training was on average based on half of them, making it the attribute with the lowest number of examples (together with `perpetrator_number`). Although ELIE could base its predictions on only few facts, its pre-classified F-measure (56%) was higher than that of the human results (48%). Even the runs on the complete corpus reached a slightly better F-measure (49%) than the human tester. Looking at the results for recall and precision individually shows that while they were almost equally good for ELIE, the human performance yielded high recall (in fact, the highest recall reached) but a significantly lower precision. For easily identifiable facts such as time information, inattentiveness during the review of the documents reduce the achieved performance measures. While people can miss some facts that they would otherwise have extracted, machines work thoroughly and deterministically.

As mentioned in the human performance analysis, the specified place attributes received low scores although their properties (geographic names) suggested that they should be easy to identify. The very low score (F-measure 7.5%) of the human tester's results for the `country` attribute (see page 25 f), that was in part a result of inattentiveness, was easy to beat. As the training examples for this attribute also contained inconsistencies, ELIE also did not do as well as it did on the other specific place attributes (`town`, `department`) that had a similar structure (geographic names).

The `location` attribute, which was part of the original target structure, was replaced by the specific geographic attributes because many reports contained more than one piece of location information. The fact that both the human tester and the machine received lower scores for

`location` than for `town`, `department`, `country` combined substantiated the decision to change the target structure.

Comparison: Operating Times

Our experiment proved the common observation to be true that machines can perform tasks faster than people. Although we have no exact data on the man-hours invested in training and fact extraction in the human experiment, they clearly exceeded the average operating times of ELIE (an average of 18.6 hours for preprocessing, training, and fact extraction for all attributes combined) severalfold.

Comparison: Semantic Extraction Quality

The human tester's incorrect extractions contained a large number of co-references to expected facts, partial matches, and other information that was part of a description of a terroristic act (see chapter 4.1 Human Performance Results) even if it was not extracted (e.g. because the report included other facts for the same attribute or it was missed during expected result preparation).

In general, the tester's extracted facts were semantically correct and within the description of a terroristic act or it was open to interpretation if they qualified as information to be extracted (reports of threats, references to previous acts that provided only little information).

Partial matches or extractions with ambiguous starts/ends usually made sense and did not change the fact's meaning when looking at them alone. For example, in "*[...] operator of a tranmission tower for channel 7 television was killed [...]*" 544) the `victim_target` extraction clearly starts with "*victim*" but may include the following tokens up to "*tower*" or "*television*", depending on the desired level of detail; ending the extraction after "*transmission*" or "*channel*" would not result in a sound extraction, neither would extractions starting/ending with "*and*", ending with "*the*", or in the middle of a word. No partial tokens were extracted.

Due to the IE guidelines we followed in the experiment, there were no overlapping extractions. Therefore no extraction contained fact descriptions for more than one attribute as those would be put into the other attribute's slot.

The number of "nonsense" extractions – words or phrases that did not describe facts of the corresponding attribute – was low.

The IE system's incorrect results also included co-references, partials, and other terroristic act facts that were not part of the expectations as well. Those were less frequent than they were in the human tester's extractions, however.

Extractions that held tokens that had appropriate parts-of-speech but were not given in the context of a terroristic act description were common, too.

There were numerous long extractions that contained more than one piece of information, including facts for different attributes. Since ELIE's algorithm and operating mode allowed overlapping extractions, parts of this type of long extraction were extracted additionally.

A couple of `action` extractions contained only parts of a word, e.g. extracting only "murder" in "*soldiers murdered 17 fishermen*" (367).

There were more "nonsense" extractions (e.g. weapon: "terrorist" (260)) than the human tester made.

Both the human tester's and the machine's results included several partially correct (thus evaluated incorrect) extractions that contained more information than what was expected. While the

human tester usually extracted more details about the expected fact itself (e.g. "*were wounded*" in the `victim_target` extraction *"11 persons, including 2 newsmen, were wounded [...]"*, document 437), the machine's additional data often included information for other attributes (`perpetrator`: "*Dr Antonio Regalado was the perpetrator of the murder of Msgr Romero*", 062) or tokens that were unrelated (`weapon` extraction: "*reports that three other car bombs*", document 223).

The IE system applied learned patterns better than the human tester because the system had only few other resources besides the training examples, while a person's own fact identification patterns counteract the acquisition and application of someone else's.

Inconsistencies usually make training more difficult as they do not provide clear patterns for the machine's data model and a person's pattern identification. In cases of single errors, human readers have the advantage that they can identify single, obviously wrong extractions for an attribute as errors (e.g. in our training examples: the date "*22 March*" extracted as `location`) and will not consider them when looking for patterns in the training examples.

Errors that occur more than once or may even have a pattern, however, can have the opposite effect. In our experiments, performance for the `country` attribute showed that the machine produced correct matches by applying the data model trained on inconsistent training examples and extracting information that was not a country description for a terroristic act (document 102: "*[...] the meeting scheduled between the government and the FMLN in Caracas, Venezuela, has been suspended*"). In contrast, the human tester used geographical knowledge and context understanding to identify slot fills, which resulted in false negatives for expected results that had been extracted by mistake.

Machine performance has the advantage of consistent data model application. It does not overlook information, gets distracted or extract information into a wrong slot by accident, which human testers may do. This was present in our experiment (e.g. many of the human tester's expected `country` extractions were simply overlooked).

The IE system's quantitative performance measures where worse than the human tester's. The machine also produced more syntactically wrong extractions.

Yet the fact that the human performance was not near-perfect demonstrated that the reports in the corpus were rather complex.

5. Conclusion

ELIE's F_1-measure of a little less than one third (25% and 31% on a pre-classified corpus) in our experiment, the performance does not seem very good. Yet the human tester's result of 41% demonstrated that the task was rather difficult. As people have a better understanding of natural language, it is not likely that a machine can achieve extraordinary performances for difficult texts.

The breakdown of results into individual attributes showed that there are big performance differences within the individual attributes to be extracted for both the machine and the human tester. The analysis suggests that the difficulty of an attribute's expected extractions and the number of examples to train from greatly influence the achieved performance.

The machine and the human tester reached only very low results for the `country` attribute, for which the the training examples and expected results contained inconsistencies and errors. The identification and extraction of a country's name should not be very difficult. This indicates that the quality of the training examples and expected results also have a big influence on performance.

Incorrect (machine) extractions that contain correct facts (e.g. partial matches that include the expected extractions, co-references, facts that were not included in the expected results by mistake) provide enough information to return correct search results in queries requesting specific information or similar IE tasks requiring a structured information extraction data collection. For example, if looking for information on terroristic acts perpetrated by a specific organization, searching for extractions containing the organization's name return more useful data on the organization's acts than those that were counted as correct in the quantitative result analysis. Looking at the other slot fills from the same document helps determine if these extractions were part of the description of a terroristic act.

As we received numerous extractions in our machine experiment that contained correct data but were not correct matches due to the characteristics described above, the data base provided by the IE system is more helpful and offers more usable data than the performance measures suggested for some applications. This data may not be useful for other employments of IE data processing, however.

With our results we demonstrated that the capabilities of IE systems and people depend on the complexity of the text and the quality of the provided training examples.

In addition to that, we found that there was more relevant data in the extractions than the performance measures suggested. While these extractions do not count towards the evaluation results, they may be good enough for queries when looking for information about facts in the data.

6. Implementation: Evaluation Module for XTract

XTract is an extensible tool for manual and automatic fact extraction and extraction analysis from natural language texts. The current version (2.06.02) provides a graphical user interface for manual fact extraction. It was developed by Heiko Kahmann [10].

For manual fact extraction, natural language text documents can be loaded into a text window. The target structure must be created in a relational database and then a graphical representation of the target structure is displayed in the XTract window. Facts from the text can be extracted and saved in the database with simple drag-and-drop gestures.

The XTract architecture uses a plug-in system for a simple addition of new functionality. Configuration information, such as the database connection data, is provided in property files.

The database schema must contain system data tables of a predefined structure in addition to the target structure table.

XTract was implemented in Java 1.4. It connects to relational databases to store and retrieve extracted data.

6.1 New Module Functionality

The new module we developed as part of this thesis adds evaluation functionality to XTract. It allows XTract users to calculate performance measures for a given set of extractions and expected results.

In addition to the quantitative performance measures, the individual extractions can be viewed in order to gain more information about them. The source documents are displayed in a graphical user interface. The existing extractions can be highlighted in order to compare the evaluated extractions to the expectations. This gives the user the opportunity to examine the extractions and learn more about their quality and error patterns.

6.2 Evaluation Module Description

The evaluation module accepts databases containing extractions made with XTract and files with fact positions as input. A corpus document list identifying which documents' extractions are to be considered and which were used for training ("shuffle") can be entered as well. Performance measure calculations using "one answer per occurrence" or "one answer per document" evaluation are possible.

Screenshot 1: Evaluation Set-up Window

The sets of extractions and the shuffle are selected in the Evaluation Set-up window. The database currently selected in XTract will be evaluated. It is also possible to evaluate the performance of external extractions instead if they are provided in a document containing their fact positions. By entering the file path in the bottom entry field, the external extraction list will be used instead of the database.

The source of the expected extractions can be a database loaded in XTract (selection list on the left) or a document with a fact positions list (middle entry field on the right).

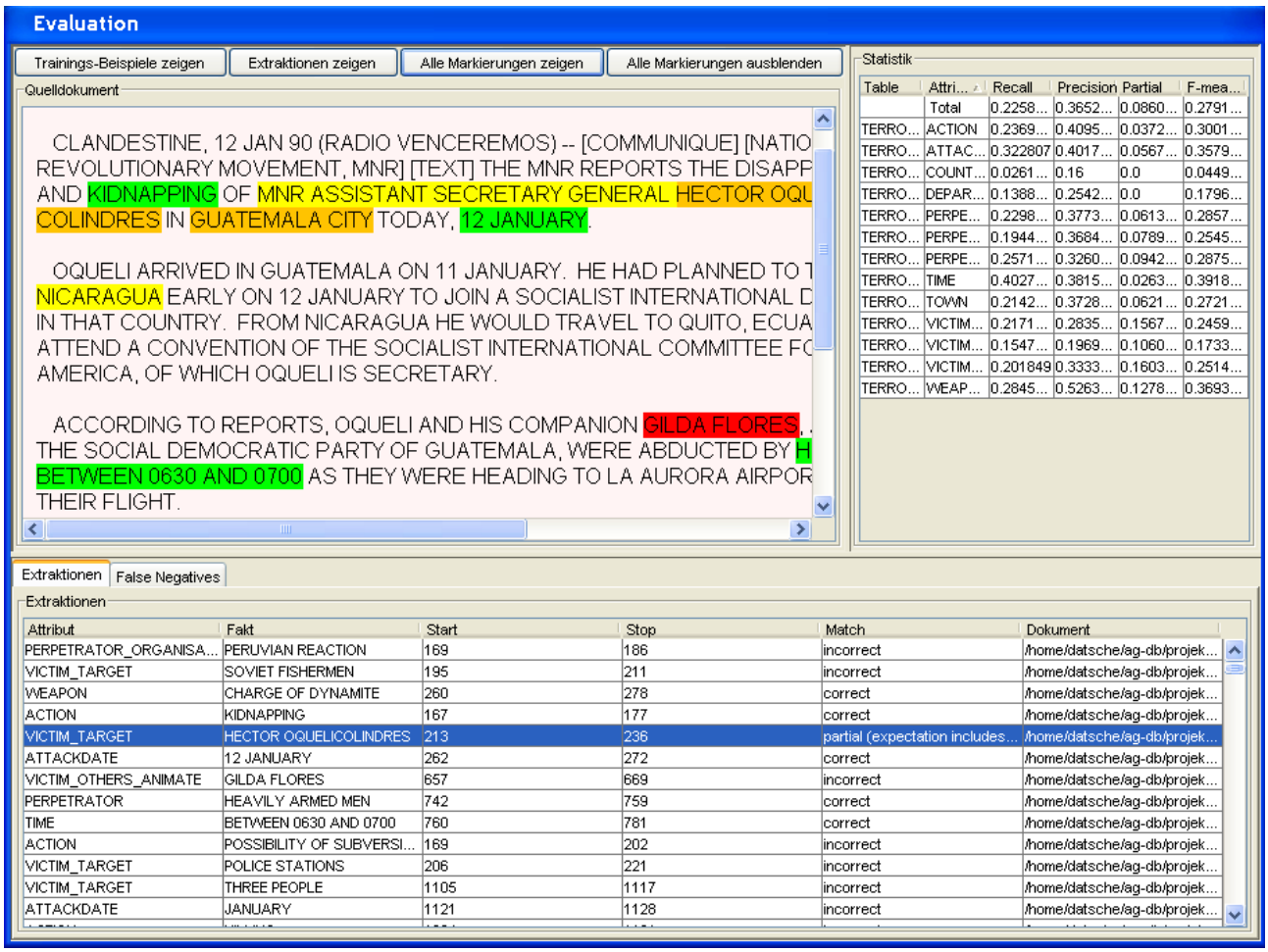
The list of documents that specifies which documents were used for training and which for fact extraction ("shuffle") can be entered in the top entry field. The fraction of training examples in the shuffle must be inserted. With the check box next to this, one can choose of all documents or only the relevant documents (i.e. those which contain extractions in the expected results) should be looked at.

The default setting for the evaluation is one answer per occurrence. Checking the check box at the bottom, one the performance measures for answer per document will be displayed in the Evaluation window.

Clicking on the “Berechnen” button will start the calculations:

1. The extractions from the selected databases or files are collected stored in HashMaps.
2. The performance measures are calculated.
3. The tables with extractions and statistics for the evaluation window are prepared

Screenshot 2: Evaluation Window



The evaluation window displays the evaluated extractions as a list and the corpus documents with extractions as highlights.

The table can be sorted by any of the columns. When one extraction is selected, the document it was extracted from is displayed in the text window and the fact is highlighted. A similar table for the false negatives is available in the second tab.

The buttons above the document display provide different highlighting of the text: extractions to be evaluated: all expected facts (yellow), all evaluated facts (correct: green, partial: orange, incorrect: red), all extractions, no highlights.

The performance measures are shown in the table on the right side. Recall, precision, F_1 F-measure, and partial matches are given for each attribute and all extractions combined.

References

[1] C. Siefkes and P. Siniakov. An Overview and Classification of Adaptive Approaches to Information Extraction. In *LNCS Journal on Data Semantics*, 2005.

[2] A. Finn and N. Kushmerick. Multi-level Boundary Classification for Information Extraction. In *Proc. European Conference on Machine Learning (Pisa)*, 2004.

[3] A. Finn and N. Kushmerick. Information Extraction by Convergent Boundary Classification. *AAAI-04 Workshop on Adaptive Text Extraction and Mining (San Jose)*, 2004.

[4] C. Will. Comparing human and machine performance for natural language information extraction: Results for English microelectronics from the MUC-5 evaluation. *Message Understanding Conference archive; Proceedings of the 5th conference on Message understanding (Baltimore, Maryland)*, 1993.

[5] G. King and W. Lowe. An automated information extraction tool for international conflict data with performance as good as human coders: A rare events evaluation design. In *International Organization Vol. 57, No. 3*, 2003.

[6] R. Gaizauskas and Y. Wilks. Information Extraction: Beyond Document Retrieval. In *Journal of Documentation 54(1)*, 1998 .

[7] N. Chinchor, L. Hirschman, and D. Lewis. Evaluating Message Understanding Systems: An Analysis of The Third Message Understanding Conference (MUC-3). In *Computational Linguistics 19(3)*, 1993.

[8] W. Lehnert, C. Cardie, D. Fisher, J. McCarthy, E. Riloff and S. Soderland. Evaluating an Information Extraction System. In *Journal of Integrated Computer-Aided Engineering*, 1994.

[9] J. Hobbs. The Generic Information Extraction System. In *Proceedings, Fifth Message Understanding Conference (MUC-5)*, 1993.

[10] H. Kahmann. Manuelle Faktenextraktion. Diplomarbeit, FU Berlin, 2004

[11] R. Grishman and B. Sundheim. Message understanding conference - 6: A brief history. In *Proceedings of the 16th International Conference on Computational Linguistics*, June 1996.

[12] B. Sundheim. Tipster/MUC-5 information extraction system evaluation. *Annual Meeting of the ACL; Proceedings of a Workshop held at Fredericksburg, Virginia: September 19-23, 1993*, 1993.