A Knowledge-Extraction Approach to Identify and Present Verbatim Quotes in Free Text

Gerhard Paaß Andre Bergholz Anja Pilz
Fraunhofer Institute Intelligent Analysis and Information Systems (IAIS)
Schloss Birlinghoven
St.Augustin, Germany
{gerhard.paass|andre.bergholz|anja.pilz}@iais.fraunhofer.de

ABSTRACT

In news stories verbatim quotes of persons play a very important role, as they carry reliable information about the opinion of that person concerning specific aspects. As thousands of new quotes are published every hour it is very difficult to keep track of them. In this paper we describe a set of algorithms to solve the knowledge management problem of identifying, storing and accessing verbatim quotes. We handle the verbatim quote task as a relation extraction problem from unstructured text. Using a workflow of knowledge extraction algorithms we provide the required features for the relation extraction algorithm. The central relation extraction procedures is trained using manually annotated documents. It turns out that structural grammatical information is able to improve the F-vale for verbatim quote detection to 84.1%, which is sufficient for many exploratory applications. We present the results in a smartphone app connected to a web server, which employs a number of algorithms like linkage to Wikipedia, topics extraction and search engine indices to provide a flexible access to the extracted verbatim quotes.

Categories and Subject Descriptors

I.2.7 [**Computing Methodologies**]: Artificial IntelligenceNatural language processing; H.3.1 [**Content Analysis and Indexing**]: Linguistic processing—*relation extraction, quote extraction*

General Terms

Natural language processing, text mining

Keywords

Relation Extraction, Information extraction application

1. INTRODUCTION

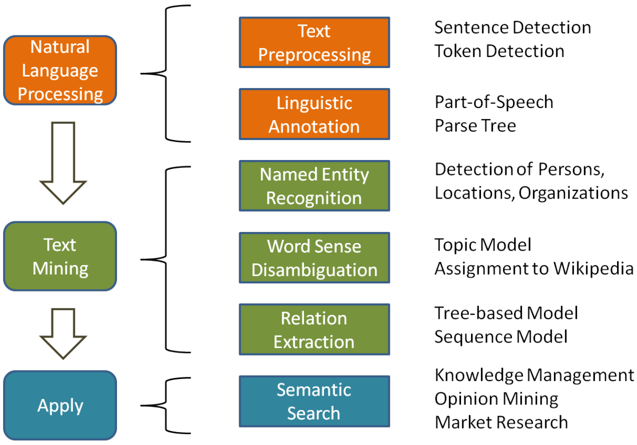

Figure 1: The workflow of information extraction tasks.

During the last decade the style of news stories and communication changed. Journalists prefer to select news stories usually according to associated conflict, deviance, negativity and impact, and very often concentrate on persons, especially on high status actors [5] such as influential politicians. In this situation it is extremely important to identify verbatim quotes of persons in news stories as these quotes have a higher probability to be correct. Note that in most countries, e.g. in Germany, there are specific press laws and supreme court decisions that require a very diligent citation of verbatim quotes.

In each hour many thousand news stories with verbatim quotes are communicated in each country. Obviously it is extremely costly to read all these news stories and extract the quotes manually. Hence there is the need for an automatic procedure for the extraction of these quotes. Subsequently they can be stored in a knowledge management system and used for applications, e.g. in press, marketing, public relations or by normal citizens which are interested in quotes of their favourite sportspersons or actors.

In this paper we describe an approach to extract verbatim quotes of persons from a text by text mining methods. We present a smartphone application which continually updates the quote database and allows the user to query the database and observe the quotes of his favourites.

2. INFORMATION EXTRACTION

Information extraction refers to the automatic extraction

iKnow '12 Sep 05-07 2012, Graz, Austria

of meaningful information from unstructured machine-readable text documents in a corpus. Examples are the annotation of phrases as *entities* (names of persons, organizations, etc.) or as *concepts* from an *ontology* (e.g. Wikipedia). Moreover *relationships* between concepts or entities may be extracted. Information extraction usually consists of a number of steps which can be grouped into *Natural Language Processing*, *Text Mining* and *Application*, as shown in figure 1.

Two different approaches may be used to arrive at automatic extraction methods. For simple tasks explicit rules may be constructed manually. For more complex problems statistical classification and clustering models are determined using training data. To describe the syntactic structure of sentences we use the Stanford Parser [2]. Figure 2 shows dependency parse trees generated by this parser. It links each word to the words that depend on it and places the dependent word in a lower level.

Named entity extraction has the task to identify names in a text and assign it to one or more categories, e.g. person, location, organization. We use *Conditional Random Fields* (CRF) [3] trained on annotated sentences to determine named entities. In earlier experiments using the CoNLL data [8] we arrived at the following F-values for German text: Persons 90.4%, locations 88.4% and organizations 78.7%.

For the extraction of verbatim quotes we require two different types of entities. We extracted persons with a CRF using the CoNLL data as well as additional manually annotated training documents. To get potential verbatim quotes we used a regular expression search for quotation marks. These marks often surround a quotation containing direct speech. However they can also be used to indicate a literal title or name, as well as a different meaning of a word or phrase than the one typically associated with it. Quotation marks are also often used to express irony or emphasis. Note that there may occur quotes within quotes which usually are expressed by different quote characters, e.g. guillemets «...». We used the pairs of international quotation marks given in [9], which also allow to detect quotations within quotes.

3. RELATION EXTRACTION AND EXPERIMENTS

Assume that by named entity recognition we have identified all persons in a document and all potential quotes delimited by quotation marks. Then the problem of estimating whether a person has uttered a verbatim quote can be considered as a relation extraction task. *Relation Extraction* deals with the problem of finding semantic associations between entities within a text phrase (i. e. usually a sentence). Given a fixed binary relationship of type r in the set R of relationships, the goal is to extract all instances of entity pairs that have the relationship r between them. More precisely given a text snippet x and two marked entities E_1 and E_2 in x, identify if there is any relationships $r \in R$ such that $r(E_1, E_2)$. The set R of relations includes an alternative relation `other` if none of the predefined relations holds.

As unsupervised approaches to relation extraction yield lower performance levels [10] we concentrated on supervised approaches. Very good results have been obtained with *kernel methods* [6] that design special kernels to capture the similarity between structures such as trees and graphs. The combination of dependency parse trees and syntactic parse trees leads to an F-score of 81% on the ACE-2003 benchmark data for the `role`-relation [7]. The approaches may also be applied to German text yielding an F-score of 77% on a newspaper corpus for the `memberOf`-relation [6].

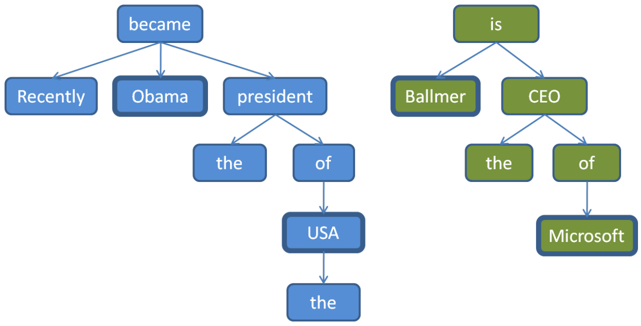

Figure 2: Dependency Parse Tree for two Sentences.

Relation extraction can also be considered in a *probabilistic modelling framework* evaluating statistical dependencies between terms. [1] extend CRFs for the extraction of semantic relations from text. In this paper we apply CRFs to relation extraction using a specific encoding of labels. This encoding takes into account that the same named entity may utter several quotes, but a single quote can only belong to a single named entity. Consider, for instance, the text snippet "We aspire to success," Obama said. "But the rich should pay fair taxes." Here the single entity 'Obama' has uttered two different verbatim quotes. To alleviate the detection of this situation for the CRF we have marked the first quote with 'A', the person uttering the quote with 'PER' and the last quote with 'E'. The text between the first quote and the person entity is annotated with 'B' and the text between the person entity and the subsequent quote is annotated with 'D'. This annotation is shown in the first example in figure 3. Note that the quote relations spans over two sentences.

Only a person entity who uttered a quote should be annotated with 'PER' and only a text in quotes which actually was said by a nearby person entity should be annotated as 'A' or 'E'. Therefore in the second example the named entity 'Warren Buffet' is annotated by 'o', as it is not involved in a verbatim quote. It is possible, that a person entity which is not involved in a quote occurs in the parts annotated as 'A', 'B', 'D', or 'E'. This excludes the annotation of overlapping quote relations, which, however, until now did not occur in our documents. Note that text in quotes which was actually not said by a person is annotated as 'o'.

The named entities extracted beforehand as well as the potential quotes between quotation marks are encoded as input features for the CRF. The labels 'A', 'B', 'PER', 'D', and 'E' are output labels which have to predicted by the model. We developed a special feature extraction language, which allows to form a large number of different features and to combine these features by conjunctions and disjunctions. We denote the extracted person entities by exPER, the text between two quotation marks by exQU, and the dependency tree by dTree. We employed the input features shown in table 1.

The features are divided into three groups: word features characterize properties of the current word, such as capitalization, multi-word features are features computed from several words, e.g. bigrams and trigrams of POS-tags or

o	A	A	A	A	A	B	PER	D	D	E	E	E	E	E	E	E	E	o
“	We	aspire	to	success	,	”	Obama	said.	“	But	the	rich	should	pay	fair	taxes	.	”

PER	D	D	D	D	D	E	E	o	o	o	o	o
Obama	calls	the	plan	the	“	Buffet	Rule	”	for	Warren	Buffet	.

Figure 3: Two examples of labels (upper row) of a sentence (lower row) for quote extraction.

the relative position of a word with respect to extracted named entities. The last group uses information from the parse trees. This allows to introduce structured syntactic knowledge into the CRF, which usually only considers the information from direct neighbours. The first feature is the dominating verb of a word inside of a named entity. It is found by going up in the dependency parse tree until you find a verb. In the left tree in figure 2, for instance, the dominating verb of both 'Obama' and 'USA' is 'became'. If there is no dominating verb then the feature is not present. The least common ancestor of two words in a tree is the lowest node which is above each of the two words. In the right tree in figure 2, for example, the least common ancestor of 'Ballmer' and 'Microsoft' is 'is'. Note that the negation 'not' has the specific POS-tag 'PTKNEG'. Using the parse tree it be assigned to the corresponding verb even if there are some intervening phrases. The same holds for the auxiliary verb 'VAFIN' or modal verb 'VMFIN' often related to a verb past participle verb 'VVPP'.

From the definition of the labels in figure 3 we know that a correct verbatim quote may have three different forms:

- A sequence A...A corresponding to a quoted text followed by zero or more B...B followed by PER...PER corresponding to an extracted named entity.
- A sequence PER...PER corresponding to an extracted named entity followed by zero or more D...D followed by a sequence E...E corresponding to a quoted text.
- A sequence A...A corresponding to a quoted text followed by zero or more B...B followed by PER...PER corresponding to an extracted named entity followed by zero or more D...D followed by a sequence E...E corresponding to a quoted text.

This defines the criterion for accepting entity-quote pairs as corresponding to a verbatim quote. Only if the A...A and E...E exactly correspond to a text delimited by quotes and if PER...PER exactly corresponding to an extracted named entity the verbatim quote is accepted. In addition there must not be a gap in the annotations B...B and D...D. In this way the detection is quite conservative which also can be seen from the high precision values.

We trained the CRF using 640 annotated news stories. The training of the model required about an hour on a commodity PC. We performed a 5-fold crossvalidation. The results are given in table 2. If only word features are used we arrive at an F-value of 79.5%. This means roughly that four of five quotes are correctly identified as a verbatim quote and are correctly assigned to the speaker. If we add the parse tree features the results improve significantly to an F-value of 84.1%. Obviously the structural information contained in the parse trees gives valuable hints on the verbatim quote relation. For our application the results are very appropriate, especially as the precision is quite high (88.8%), and

Table 1: Input features for the conditional random field for quote extraction.

Word features: mark the current word with ...
the word string, its lemma and POS-tag.
3-character suffix of the word
4-character suffix of the word.
word shapes: all lower case, all upper case, etc.
which is a 'finite verb' with word string and POS tag.
which is an 'auxiliary verb' with word string and POS tag.
which is a 'modal verb' with word string and POS tag.
beginning and end of sentences.
Multi-Word features: mark the current word with ...
POS(previous word)_ POS(current word).
POS(previous word)_POS(current word)_POS(next word).
if it is between a pair exPER and exQU.
if it is between a pair exQU and exPER.
if it is in an exPER.
if it is in an exQU.
with its extracted chunk phrase.
Parse-tree features: mark each word ...
in an exPER with the dominating verb in dTree.
in an exQU with the dominating verb in dTree.
in an exQu with the least common ancestor with the previous exPER in dTree
in an exQu with the least common ancestor with the following exPER in dTree
which is a finite verb with the corresponding negation.
which is a past participle with the corresp. auxiliary verb.
which is a verb with the corresponding particle, e.g. "go to".

only about one of ten proposed verbatim quotes is wrong. Note that this performance rate also covers errors in named entity recognition.

4. THE QUOTE SMARTPHONE APP

Extracted quotes are stored in a knowledge management system and may be retrieved for applications, e.g. by press, marketing, or public relations professionals or by normal citizens which are interested in quotes of their favourite sportspersons or singers. Figure 4 shows a smartphone app for browsing the quote extraction results. The extraction of verbatim quotes resides on a server. It processes more than 100 newsfeeds continuously collecting the latest news stories. These stories are then propagated to the extraction pipeline (figure 1) where person named entities and potential quotes are identified and finally the verbatim quotes are annotated. In addition the topics addressed in the full article are identified by a topic model and added to the quote in the form of hash tags (e.g. #athen). Using LDA topic models and the disambiguation algorithm described in [4] the named entity corresponding to a verbatim quote is disambiguated and linked to Wikipedia. If the associated Wikipedia article contains a picture of the person, this picture is presented in the GUI. As often the same quote appears in different publications, we also have included a deduplication step.

On startup the app presents a list of recent quotes. By

Table 2: Results for different experiments

Experiment	Prec.	Recall	F-val (std err)
word features	85.8	74.2	79.5 (1.2)
word + parsetree features	88.8	79.9	84.1 (0.9)

clicking the quote the user can switch to the corresponding news story where the quote was found. To get comprehensive access to quotes and corresponding articles they are indexed in full text search indices and the user can retrieve quotes and articles containing specific text fragments. An alternative access is provided by a tag cloud representing important tags as well as by a list of all quoted persons. Finally the user can select favourite persons to follow their quotes more closely. The app runs on any browser and is especially adapted to current smartphones. It was successfully presented on the CeBIT computer fair 2012 to a larger audience.

5. CONCLUSION

In this paper we described a set of algorithms to solve a knowledge management problem. We considered the verbatim quote problem and solved it as a relation extraction problem from text. Using a workflow of knowledge extraction algorithms we provides the required features for the relation extraction algorithm. It turns out that structural grammatical information is able to improve the F-vale for verbatim quote detection to 84.1%, which is sufficient for many exploratory application. We present the results in a smartphone app connected to a web server, which assembles a number of algorithms like linkage to Wikipedia, topics extraction and search engine indices to provide a flexible access to the extracted verbatim quotes.

The approach can be generalized directly to other relevant knowledge management task, e.g. to identify if a person is employed by a specific company (compare [7]) or is an expert in a specific field. A drawback of the approach is the need to provide training documents annotated with the target relation. Currently there are many efforts to reduce this effort and arrive at weakly supervised relation extraction approaches [10]. Although currently the performance of these methods is usually not sufficient, the field is rapidly evolving and has a potential for large improvements.

Acknowledgments

The work presented here was funded by the German Federal Ministry of Economy and Technology (BMWi) under the THESEUS project.

6. REFERENCES

[1] M. Bundschus, M. Dejori, M. Stetter, V. Tresp, and H.-P. Kriegel. Extraction of semantic biomedical relations from text using conditional random fields. *BMC Bioinformatics*, 9, 2008.

[2] M.-C. de Marneffe, B. MacCartney, and C. D. Manning. Generating typed dependency parses from phrase structure parses. In *Proc. LREC 2006*, pages 449–454., 2006.

[3] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. ICML*, pages 282–289, 2001.

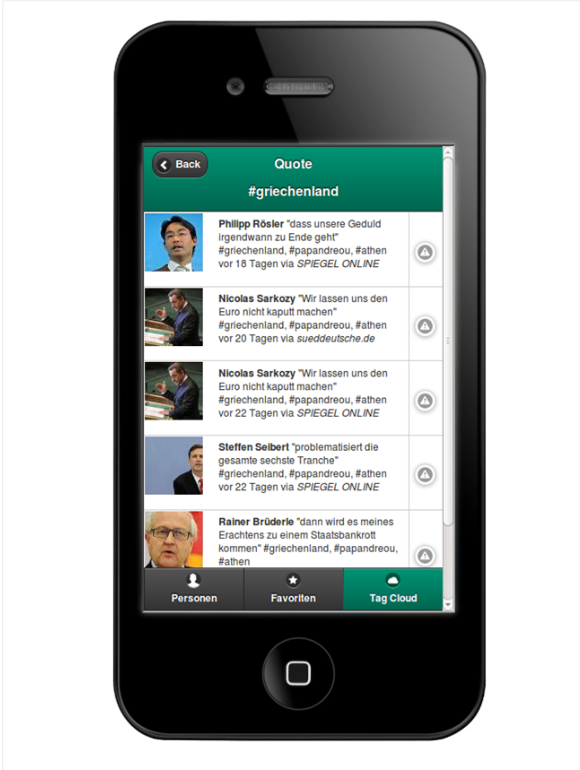

Figure 4: The smartphone app for browsing the quote extraction results.

[4] A. Pilz and G. Paass. From names to entities using thematic context distance. In *CIKM 2011*, 2011.

[5] T. Rantanen. *The Media and Globalization*. Sage, London, 2004.

[6] F. Reichartz. *Automatic Relation Extraction from Text*. PhD thesis, University of Bonn, 2011.

[7] F. Reichartz, H. Korte, and G. Paass. Semantic relation extraction with kernels over typed dependency trees. In *Proc. ACM SIGKDD*, pages 773–782, 2010.

[8] E. F. T. K. Sang and F. D. Meulder. Introduction to the conll-2003 shared task: language-independent named entity recognition. In *Proc. HLT-NAACL 2003*, pages 142–147, Morristown, NJ, USA, 2003. ACL.

[9] Wikipedia-Quote. Anführungszeichen. Article in German Wikipedia, retrieved on April 1., 2012, 2012.

[10] A. Yates and O. Etzioni. Unsupervised methods for determining object and relation synonyms on the web. *J. Artif. Intell. Res. (JAIR)*, 34:255–296, 2009.