KRAKEN: N-ary Facts in Open Information Extraction

Alan Akbik **Alexander Löser**
Technische Univeristät Berlin
Databases and Information Systems Group
Einsteinufer 17, 10587 Berlin, Germany
alan.akbik@tu-berlin.de, aloeser@cs.tu-berlin.de

Abstract

Current techniques for Open Information Extraction (OIE) focus on the extraction of binary facts and suffer significant quality loss for the task of extracting higher order N-ary facts. This quality loss may not only affect the correctness, but also the completeness of an extracted fact. We present KRAKEN, an OIE system specifically designed to capture N-ary facts, as well as the results of an experimental study on extracting facts from Web text in which we examine the issue of fact completeness. Our preliminary experiments indicate that KRAKEN is a high precision OIE approach that captures more facts per sentence at greater completeness than existing OIE approaches, but is vulnerable to noisy and ungrammatical text.

1 Introduction

For the task of fact extraction from billions of Web pages the method of Open Information Extraction (OIE) (Fader et al., 2011) trains domain-independent extractors. This important characteristic enables a potential application of OIE for even very large corpora, such as the Web. Existing approaches for OIE, such as REVERB (Fader et al., 2011), WOE (Wu and Weld, 2010) or WANDERLUST (Akbik and Bross, 2009) focus on the extraction of binary facts, e.g. facts that consist of only two arguments, as well as a fact phrase which denotes the nature of the relationship between the arguments. However, a recent analysis of OIE based on Semantic Role Labeling (Christensen et al., 2011) revealed that N-ary facts (facts that connect more than two arguments) were present in 40% of surveyed English sentences. Worse, the analyses performed in (Fader et al., 2011) and (Akbik and Bross, 2009) show that incorrect handling of N-ary facts leads to extraction errors, such as incomplete, uninformative or erroneous facts. Our first example illustrates the case of *a significant information loss*:

a) *In the 2002 film Bubba Ho-tep, Elvis lives in a nursing home.*
REVERB: LivesIn(Elvis, nursing home)

In this case, the OIE system ignores the significant contextual information in the argument *the 2002 film Bubba Ho-tep*, which denotes the domain in which the fact LivesIn(Elvis, nursing home) is true. As a result, and by itself, the extracted fact is false. The next example shows a binary fact from a sentence that de-facto expresses an N-ary fact.

b) *Elvis moved to Memphis in 1948.*
REVERB: MovedTo(Elvis, Memphis)
WANDERLUST: MovedIn(Elvis, 1948)

Contrary to the previous example, the OIE systems extracted two binary facts that are *not false, but incomplete*, as the interaction between all three entities in this sentence can only be adequately modeled using an ternary fact. The fact MovedIn(Elvis, 1948) for example misses an important aspect, namely the *location* Elvis moved to in 1948. Therefore, each of these two facts is an example of important, but not crucial information loss.

Unfortunately, current OIE systems are not designed to capture the complete set of arguments for

each fact phrase within a sentence and to link arguments into an N-ary fact. We view intra-sentence fact completeness as a major measure of data quality. Following existing work from (Galhardas et al., 2001) complete factual data is a key for advanced data cleansing tasks, such as fact de-duplication, object resolution across N-ary facts, semantic fact interpretation and corpus wide fact aggregation. Therefore we argue that complete facts may serve a human reader or an advanced data cleansing approach as additional clue for interpreting and validating the fact. In order to investigate the need and feasibility for N-ary OIE we have performed the following, the results of which we present in this paper:
1. We introduce the OIE system KRAKEN, which has been built *specifically* for capturing complete facts from sentences and is capable of extracing unary, binary and higher order N-ary facts.
2. We examine intra sentence fact correctness (true/false) and fact completeness for KRAKEN and REVERB on the corpus of (Fader et al., 2011).

In the rest of the paper we review earlier work and outline KRAKEN, our method for extracting N-ary facts and contextual information. Next, we describe our experiments and end with conclusions.

2 KRAKEN

We introduce KRAKEN, an N-ary OIE fact extraction system for facts of arbitrary arity.

2.1 Previous Work

Binary-OIE: Our previous system WANDERLUST (Akbik and Bross, 2009) operates using a typed dependency-style grammar representation called Link Grammar. The system traverses paths of typed dependencies (referred to as linkpaths) to find pairs of arguments connected by a valid grammatical relationship. We identified a set of 46 common linkpaths that can be used for fact extraction. Later, the authors (Wu and Weld, 2010) trained extractors in a system called WOE, one using only shallow syntactic features and one (called WOEPARSE) that also uses typed dependencies as features. The latter system learned more than 15.000 patterns over typed dependencies. In their evaluation they showed that using deep syntactic parsing improves the precision of their system, however at a high cost in extraction speed. The OIE system REVERB (Fader et al., 2011) by contrast uses a fast shallow syntax parser for labeling sentences and applies syntactic and a lexical constraints for identifying binary facts. However, the shallow syntactic analysis limits the capability of REVERB of extracting higher order N-ary facts.
Higher order fact extraction for Wikipedia: In previous work on higher order fact extraction, the focus was placed on specific types of arguments. The authors of (Hoffart et al., 2011) for example extract temporal, spatial and category information from Wikipedia info boxes. (Weikum et al., 2011) and (Ling and Weld, 2010) focused on N-ary fact types from English sentences that contain at least one temporal argument. In contrast, KRAKEN extracts N-ary facts with arbitrary argument types.

path	head of
`nsubj-↓`	subject
`nsubjpass-↓`	subject (passive)
`rcmod-↑,appos-↑`	subject (relative clause)
`partmod-↑-nsubj-↓`	subject
`dobj-↓`	object
`prep-↓, pobj-↓`	object
`prep-↓, npadvmod-↓`	object
`advmod-↓`	context (usually modal)
`tmod-↑`	context (temporal)
`parataxis-↓,nsubj-↓`	context
`ccomp-↓,nsubj-↓`	context

Table 1: Common type-paths and the type of argument head they find.

2.2 Algorithm Outline

KRAKEN expects as input a Stanford dependency parsed sentence, in which two words are *linked* if connected via a typed dependency. Each typed dependency has a *type* denoting the grammatical nature of the link, and is directed, either upward (from child to parent) or downward (from parent to child). Given such a parse, KRAKEN executes the following three steps:
1. Fact phrase detection: The system identifies a fact phrase as a chain of verbs, modifiers and/or prepositions, linked by any of the following types: `aux`, `cop`, `xcomp`, `acomp`, `prt` or `auxpass`. Examples of such chains are *has been known* or *claims to be*. A detected fact phrase may consist of only one word if it is POS-tagged as verb and not linked with any of the aforementioned types.

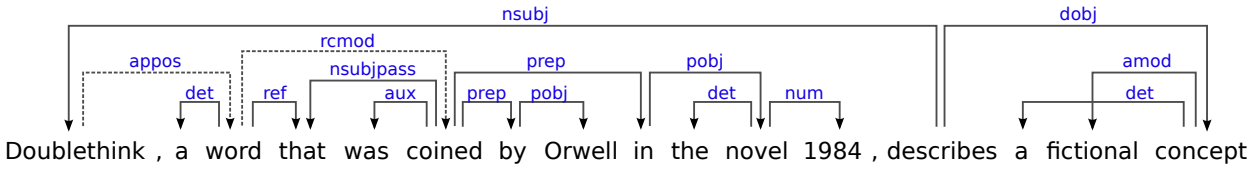

Figure 1: Example of a sentence in Stanford typed dependency formalism. One fact phrase is *was coined*. Using the type-path `rcmod-↑-appos-↑`, the subject the *Doublethink* is found, the path is highlighted in dotted lines. Using `prep-↓`, `pobj-↓`, two arguments are found: *Orwell* and *the novel 1984*. One N-ary fact for this sentence is WasCoined(Doublethink, (by) Orwell, (in) the novel 1984). The other is Describes(Doublethink, fictional concept).

2. Detection of argument heads: Next, for each word of a fact phrase, KRAKEN attempts to find heads of arguments using *type-paths* as listed in Table 1. Each type-path indicates one or more links, as well as the direction of each link, to follow to find an argument head. For example, the type-path `subj-↓` indicates that if one downward link of type `subj` exists, then the target of that link is an argument head. Figure 1 illustrates an example. At the end of this step, KRAKEN returns all found argument heads for the fact phrase.

3. Detection of full arguments: KRAKEN recursively follows all downward links from the argument head to get the full argument, excluding any links that were part of the type-path to the argument head. The combination of the detected fact phrase from step 1 and these full arguments form the fact. If a fact phrase has at least one argument, the system extracts it as a fact.

The ruleset was generated by joining the linkpaths reported in (Akbik and Bross, 2009) that contain at least one overlapping entity and one overlapping verb, and exchanging the underlying grammatical formalism with Stanford typed dependencies[1], resulting in a verb-centric and human-readable ruleset.

3 Preliminary Experimental Study

We compare REVERB, the state-of-the-art in binary fact extraction, with KRAKEN, in order to measure the effect of using N-ary fact extraction over purely binary extractors on overall precision and completeness. Additionally, we test in how far using an IE approach based on deep syntactic parsing can be used for sentences from the Web, which have a higher chance of being ungrammatical or noisy.

[1] http://nlp.stanford.edu/software/dependencies

3.1 Experimental Setup

Data set: We use the data set from (Fader et al., 2011) which consists of 500 sentences sampled from the Web using Yahoo's random link service.[2] The sentences were labeled both with facts found with KRAKEN and the current version of REVERB.[3] We then paired facts for the same sentence that overlap in at least one of the fact phrase words, in order to present to the judges two different versions of the same fact - often one binary (REVERB) and one N-ary (KRAKEN).

Measurements/Instructions: Given a sentence and a fact (or fact-pair), we asked two human judges to label each fact as either 1) true and complete, 2) true and incomplete, or 3) false. *True and incomplete facts* either lack contextual information in the form of arguments that were present in the sentence, or contain underspecified arguments, but are nevertheless valid statements in themselves (see our examples in Section 1). In previous evaluations, such

[2] http://random.yahoo.com/bin/ryl

[3] available at http://reverb.cs.washington.edu/

	KRAKEN	REVERB		
sentences	500	500		
skipped	155	**0**		
elapsed time	319.067ms	**13.147ms**		
min. confidence	-	0	0.1	0.2
total facts	572	**736**	528	457
per sentence	**1.66**	1.47	1.06	0.91
true, complete	**308**	166	146	127
true, incomplete	**81**	256	193	162
false	183	314	189	**168**
precision	**0.68**	0.61	0.64	0.63
completeness	**0.79**	0.39	0.43	0.44

Table 2: The results of the comparative evaluation. KRAKEN nearly doubles the amount of recognized complete and true facts.

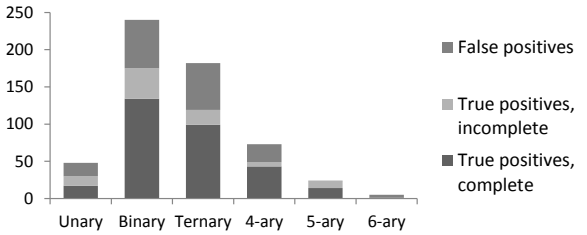

Figure 2: Distribution of arity of facts found by KRAKEN and their correctness.

facts have been counted as true. We distinguish them from *true and complete facts* that capture all relevant arguments as given by the sentence they were extracted from. We measured an inter-annotator agreement of 87%, differently evaluated facts were discussed by the judges and resolved. Most disagreement was caused by facts with underspecified arguments, labeled as false by one judge and as true and incomplete by the other.

3.2 Evaluation Results and Discussion

KRAKEN extracts higher order N-ary facts. Table 2 show results for KRAKEN and REVERB. We measured results for REVERB with different confidence thresholds. In all measurements, we observe a significantly higher number of true and complete facts for KRAKEN, as well as both a higher overall precision and number of facts extracted per sentence. The *completeness*, measured as the ratio of complete facts over all true facts, is also significantly higher for KRAKEN. Figure 2 breaks down the fact arity. KRAKEN performs particularly well for binary, ternary and 4-ary facts, which are also most common. We conclude that even though our ruleset was generated on a different domain (Wikipedia text), it generalizes well to the Web domain.

Dependency parsing of Web text. One major drawback of the settings we used is our (possibly too crude) heuristic for detecting erroneous dependency parses: We set KRAKEN to extract facts from all sentences in which the dependency parse does not contain the typed dependency `dep`, which indicates unclear grammatical relationships. A total of 155 sentences - 31% of the overall evaluation set - were skipped as a consequence. Also, the elapsed time of the fact extraction process was more than one order of magnitude longer than REVERB, possibly limiting the ability of the system to scale to very large collections of documents.

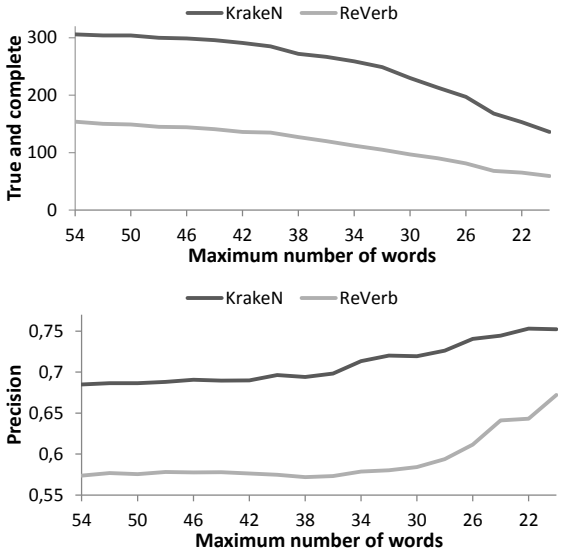

Figure 3: Impact of limiting the maximum sentence length on precision and the number of true positives.

Measurements over different sentence lengths. When limiting the maximum number of words allowed per sentence, we note modest gains in precision and losses in complete positives in both systems, see Figure 3. KRAKEN performs well even on long sentences, extracting more true and complete positives at a high precision.

Lessons learned. Based on these observations, we reach the conclusion that given the 'right portion' of sentences from a collection such as the Web, our method for N-ary OIE can be very effective, extracting more complete facts with a high precision and fact-per-sentence rate. Sentences that are well suited for our algorithm must fulfill the following *desiderata*: 1) They are noise free and grammatically correct, so there is a high chance for a correct parse. 2) They are fact-rich, so that processing resources are wisely used.

4 Summary and Future Work

Current OIE systems do not perform well for the task of higher order N-ary fact extraction. We presented KRAKEN, an algorithm that finds these facts with high precision, completeness, and fact-per-sentence rate. However, we also note that relying on a dependency parser comes at the cost of

speed and recall, as many sentences were skipped due to our heuristic of detecting erroneous parses.

Future work focuses on scaling the system up for use on a large Web corpus and increasing the system's recall. To achieve this, we will work on a first step of identifying grammatical and fact-rich sentences before applying dependency parsing in a second step, filtering out all sentences that do not meet the desiderata stated in Section 3. We intend to evaluate using very fast dependency parsers, some more than two orders of magnitude faster than the Stanford parser (Cer et al., 2010), one prominent example of which is the MALTparser (Nivre et al., 2007).

Additionally, we will examine more data-driven approaches for identifying fact phrases and arguments in order to maximize the system's recall. We intend to use such an approach to train KRAKEN for use on other languages such as German.

One interesting aspect of future work is the canonicalization of the fact phrases and arguments given very large collections of extracted facts. Unsupervised approaches that make use of redundancy such as (Bollegala et al., 2010) or (Yates and Etzioni, 2007) may help cluster similar fact phrases or arguments. A related possibility is the integration of facts into an existing knowledge base, using methods such as distant supervision (Mintz et al., 2009). We believe that combining OIE with a method for fact phrase canonicalization will allow us to better evaluate the system in terms of precision/recall and usefulness in the future.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments.

Alan Akbik received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement no ICT-2009-4-1 270137 'Scalable Preservation Environments' (SCAPE) and Alexander Löser receives funding from the Federal Ministry of Economics and Technology (BMWi) under grant agreement "01MD11014A, 'MIA-Marktplatz für Informationen und Analysen' (MIA)".

References

Alan Akbik and Jürgen Bross. 2009. Wanderlust: Extracting semantic relations from natural language text using dependency grammar patterns. In *1st. Workshop on Semantic Search at 18th. WWWW Conference*.

D.T. Bollegala, Y. Matsuo, and M. Ishizuka. 2010. Relational duality: Unsupervised extraction of semantic relations between entities on the web. In *Proceedings of the 19th international conference on World wide web*, pages 151–160. ACM.

D. Cer, M.C. de Marneffe, D. Jurafsky, and C.D. Manning. 2010. Parsing to stanford dependencies: Trade-offs between speed and accuracy. In *Proceedings of LREC*, pages 1628–1632.

Janara Christensen, Mausam, Stephen Soderland, and Oren Etzioni. 2011. An analysis of open information extraction based on semantic role labeling. In *K-CAP*, pages 113–120.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *EMNLP*, pages 1535–1545.

Helena Galhardas, Daniela Florescu, Dennis Shasha, Eric Simon, and Cristian-Augustin Saita. 2001. Declarative data cleaning: Language, model, and algorithms. In *VLDB*, pages 371–380.

Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, Edwin Lewis-Kelham, Gerard de Melo, and Gerhard Weikum. 2011. Yago2: exploring and querying world knowledge in time, space, context, and many languages. In *Proceedings of the 20th international conference companion on World wide web*, WWW '11, pages 229–232, New York, NY, USA. ACM.

Xiao Ling and Daniel S. Weld. 2010. Temporal information extraction. In *24th. AAAI Conference on Artificial Intelligence*.

M. Mintz, S. Bills, R. Snow, and D. Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.

J. Nivre, J. Hall, J. Nilsson, A. Chanev, G. Eryigit, S. Kübler, S. Marinov, and E. Marsi. 2007. Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(02):95–135.

Gerhard Weikum, Nikos Ntarmos, Marc Spaniol, Peter Triantafillou, András A. Benczúr, Scott Kirkpatrick, Philippe Rigaux, and Mark Williamson. 2011. Longitudinal analytics on web archive data: It's about time! In *CIDR*, pages 199–202.

F. Wu and D.S. Weld. 2010. Open information extraction using wikipedia. In *ACL*, pages 118–127.

A. Yates and O. Etzioni. 2007. Unsupervised resolution of objects and relations on the web. In *Proceedings of NAACL HLT*, pages 121–130.