Open Relation Extraction for Support Passage Retrieval: Merit and Open Issues

Amina Kadry
Dymatrix Consulting
Stuttgart, Germany
a.kadry@dymatrix.de

Laura Dietz
University of New Hampshire
Durham, NH, USA
dietz@cs.unh.edu

ABSTRACT

Our goal is to complement an entity ranking with human-readable explanations of how those retrieved entities are connected to the information need. Relation extraction technology should aid in finding such support passages, especially in combination with entities and query terms. This work explores how the current state of the art in unsupervised relation extraction (OpenIE) contributes to a solution for the task, assessing potential, limitations, and avenues for further investigation.

ACM Reference format:
Amina Kadry and Laura Dietz. 2017. Open Relation Extraction for Support Passage Retrieval: Merit and Open Issues. In *Proceedings of SIGIR '17, Shinjuku, Tokyo, Japan, August 07-11, 2017,* 4 pages.
DOI: http://dx.doi.org/10.1145/3077136.3080744

1 INTRODUCTION

It seems obvious that technology for extracting the meaning of text, such as relation extraction, should lead to better text retrieval methods. Yet, so far successes have been rare. This paper studies different ways of exploiting open relation extraction technology, assesses the potential for merit as well as open issues that inhibit further success for text-centric information retrieval.

Given sentences as input, open relation extraction (OpenIE) algorithms extract information on how knowledge base entities are related by analyzing the grammatical structure of each sentence.

To assess opportunities for future merit, we choose a text ranking task that operates on the sentence level and for which information about entities and relations is clearly pertinent: Retrieving explanations for how/why a knowledge base entity is relevant for an information need. This task is useful whenever entities are displayed along with web search results, such as entity cards [3].

Task (support passage ranking): A user enters information need Q; an external system predicts a ranking of relevant entities $\mathcal{E}$. Our task is to, for every relevant entity $e_i \in \mathcal{E}$, retrieve and rank K passages s_{ik} that explain why this entity e_i is relevant for Q.

We postulate and study the following hypothesis. For a given entity e_i, passages s_{ik} that explain a relevant relationship involving the entity e_i, are also good human-readable descriptions of *why* the entity is relevant for the information need Q.

Of course, conventional OpenIE algorithms have no knowledge of the information need Q. Therefore, we study outcomes of relation extraction in superposition with retrieval models such as query likelihood. This paper studies how much OpenIE contributes to accomplishing this task. While there are many suggested approaches to OpenIE, we focus on the ClausIE system, which has been shown to be one of the best OpenIE methods on three established benchmark datasets [5].

Contributions. This paper features an in-depth study of the utility of a state-of-the-art OpenIE extraction system. We study how relation extraction can help, what are promising avenues for further research, and what are limitation of current relation extraction approaches that need to be overcome.

We demonstrate that OpenIE methods provide significantly better indicators for entity-centric passage ranking tasks, in contrast to low-level NLP methods such as part-of-speech tagging, named entity recognition, or dependency parsing. Despite these significant improvements, we quantify how limitations of current OpenIE systems are affecting the quality of downstream information retrieval tasks.

Outline. The state-of-the-art is summarized in Section 2. A short introduction to the relation extraction system ClausIE is given in Section 3. Section 4 details the feature-based learning-to-rank approach through which we evaluate the merit of OpenIE technology. Quantitative experimental results are provided in Section 5.

2 RELATED WORK

Relations and retrieval. Given a relationship in a knowledge graph, Voskarides et al. [14] study the problem of finding human readable descriptions of that relationship. The relationship is given in the form $\langle e_i, r, e_j \rangle$, where e_i and e_j are given entities, i.e., nodes in the knowledge graph and r is a type of a relationship, such as *works_for*. Given this relationship, the task is to rank text passages s_{ijk} by how well they describe the relationship in human-readable form. This is the inverse problem to relation extraction [5, 11] where the task is to, given a textual description s_{ijk}, extract relational facts in the form $\langle e_i, r, e_j \rangle$. None of these approaches take a further information need Q into consideration.

In the context of web queries, Schuhmacher et al. [12] apply supervised relation extraction to documents that are relevant for the information need Q and study how many of the extracted relations $\langle e_i, r, e_j \rangle$ are indeed relevant for Q. They also analyze sentences, such as s_{ijk}, from which the relevant relation were extracted.

Sentence retrieval. Previous work on retrieving entities and support sentences addresses the sentence retrieval problem. For

SIGIR '17, August 7-11, 2017, Shinjuku, Tokyo, Japan

978-1-4503-5022-8/17/08…$15.00
DOI: http://dx.doi.org/10.1145/3077136.3080744

example Blanco et al. [4] present a model that ranks entity support sentences with learning-to-rank. Their work focuses on features based on named entity recognition (NER) in combination term-based retrieval models. Many features based on using knowledge graph entities for text retrieval could also be applied here, such as the latent entity space model of Liu et al. [9].

Temporal event summarization. Temporal summarization is the task of identifying short and relevant sentences about a developing news event such as disasters, accidents, etc. [1] in a real-time setting. Each event can be seen as a textual query that describes the event. For example, Kedzie et al. [8] propose to cluster sentences with salience predictions in the context of a named event within a multi-document summarization system. In line with many feature-based approaches, their system exploits term-based retrieval, query expansion, geographical and temporal relevance features.

Question answering. Given a question in natural language, Question Answering methods focus on providing correct and precise answers [13]. QA systems first use IR techniques are used to retrieve passages that contain the answer. Next these are analyzed to extract a concise answer. Whenever the question includes an entity, a solution to our task is also applicable to the first stage of question answering.

3 FOUNDATION: CLAUSIE

ClausIE [5] is an OpenIE (unsupervised relation extraction) system designed for high-precision extractions. In contrast to previous OpenIE approaches, such as TextRunner [2] and Reverb [7], ClausIE distinguishes between the discovery of useful information from a given sentence and the representation of this information through multiple propositions. The system identifies different types of clauses, such as adverbial, complement, indirect object, and direct object. In contrast to many earlier approaches, ClausIE does not require labeled or unlabeled training data or global post-processing, making it applicable to open-domain retrieval tasks.

Example. Given the following sentence with token indices:[1]
"The$_1$ rules$_2$ of$_3$ golf$_4$ are$_5$ a$_6$ standard$_7$ set$_8$ of$_9$ regulations$_{10}$ and$_{11}$ procedures$_{12}$ by$_{13}$ which$_{14}$ the$_{15}$ sport$_{16}$ of$_{17}$ golf$_{18}$ should$_{19}$ be$_{20}$ played$_{21}$."

Phase 1. Clause types are extracted, representing constituents by their head word with token offset. For example:

Complementary clause	*SVC(C:set$_8$, V:are$_5$, S:rules$_2$, A?:of$_9$)*
Adverbial clause	*SVA(V:played$_{21}$, S:sport$_{16}$, S:by$_{13}$)*

Phase 2. Propositions of relation tuples are derived. For example:

The rules$_2$ of golf	*are$_5$*	*a standard set$_8$ of$_9$ regulations*
The rules$_2$ of golf	*are$_5$*	*a standard set$_8$ of$_9$ procedures*
The rules$_2$ of golf	*are$_5$*	*a standard set$_8$*
the sport$_{16}$ of golf	*should be played$_{21}$*	*by$_{13}$ a standard set$_8$ of regulations*
the sport$_{16}$ of golf	*should be played$_{21}$*	*by$_{13}$ a standard set$_8$ of procedures*

Whenever entity and query terms are contained in the same proposition, this sentence is likely to explain the connection between query and entity.

4 APPROACH: RANKING SENTENCES FOR EXPLAINING ENTITY RELEVANCE

To study the utility of ClausIE for the support passage ranking task, we make use of a common two-step approach of 1) extracting candidate sentences and 2) using learning to rank (LTR) with a rich set of features, some of which are based on ClausIE's extractions.

4.1 Extracting Candidate Sentences

In order to create a set of candidate sentences for a given query Q and entity e_i, a corpus of documents that is pertinent to the entity is required. Any corpus could be used here, such as the ClueWeb corpus with entity links, as used by Schuhmacher et al. [12]. Assuming that OpenIE works best on grammatically well-formed sentences, we instead follow Voskarides et al. [14] and base this study on sentences from the Wikipedia article of the entity e_i.

4.2 Machine Learning (LTR)

Sentences are ranked with a list-wise learning-to-rank (LTR) approach implemented in RankLib.[2] The weight parameter is learned by optimizing for the Mean-Average Precision metric (MAP) using coordinate ascent and 20 restarts. The LTR will learn a weighted feature combination to achieve the best possible ranking on the training set. Features of different categories are discussed below. We study feature sets for their merit by applying LTR on hold-out test data using cross-validation.

4.3 Sentence Ranking Features

Table 1 details the features which fall into these categories:
Text features and quality features (Text) (1–8) capture the relevance and quality of the sentence at the term level.
NLP features (9–16) are derived from part-of-speech (POS) and named entity recognition (NER) tags. These have been speculated to not help IR.
Dependency parse tree (DP) features (17–19) capture the grammatical structure of the sentence. We use the Stanford dependency parser [10] which is also used by the ClausIE system. Earlier works on relation extraction use the direct path between two entities in the dependency parse tree [15].
ClausIE features (20–43) capture the sentence's relation information about entity and query terms. Features are divided by positions of the relation proposition, i.e., subject, verb, and object. Relation quality indicators are included, such as the proposition length measured in tokens or the maximum constituent length (number of tokens in dependency subtree)—both averaged across all propositions extracted from this sentence.

5 EXPERIMENTAL EVALUATION

We conduct a series of experiments to determine the utility and issues of an available state-of-the-art OpenIE system. We focus on the task of ranking support sentences by how well they explain the relevance of a given entity e_i for a given information need Q.

The study is divided according to three questions: 1) Under ideal conditions, could relation extractions help rank relevant passages? 2) What quality is achieved by a fully-automatic learning-to-rank

[1] See demo at https://gate.d5.mpi-inf.mpg.de/ClausIEGate/ClausIEGate.

[2] lemurproject.org/ranklib.php

Table 1: Features used for support passage ranking.

Feat.	Description
Text	
1	sentence length measured in number of words
2	sentence position measured as a fraction of the document
3	fraction words that are stop words
4	fraction of query terms covered by sentence
5	sum of ISF of query terms (ISF is inverse sentence frequency)
6	average of ISF of query terms
7	sum of TF-ISF of query terms
8	number of entities mentioned
NLP	
9-12	for nouns/verbs/adjectives/adverbs: fraction of words with POS tag
13	whether sentence contains a named entity
14-16	for NER types PER/LOC/ORG: whether NER of type is contained
DP	
17	number of edges on the path between two entities in dependency tree
18	indicator whether path goes through root node
19	indicator whether path goes through query term
ClausIE	
20	whether ClausIE generated an extraction from this sentence
21-27	for all seven clause types: whether clause of this type is extracted
28	proposition length measured in tokens
29	maximum constituent length (size of dependency tree) in proposition
30-32	for subject/object/both: if another entity is in subject and/or object position of the proposition
33-34	for subject/object position: if given entity is in position of proposition
35-36	for subject/object position: if any entity is in position of proposition
37-38	for subject/object position: if an entity link is in position of prop.
39-41	for subject/verb/object position: if a query term (ignoring stopwords) is in position of proposition
42-43	for subject/object position: if a named entity (NER) is in position of proposition

approach with OpenIE features (cf. Section 4)? 3) Which open issues of OpenIE systems inhibit the application to text ranking tasks?

5.1 Test collection

For this study we build a test collection[3] for 95 support passage rankings (one per query and entity). We use a subset of ten 2013/2014 TREC Web track queries and (up to) ten relevant entities $\mathcal{E}$ for these topics, which are taken from the REWQ gold standard.[4] To focus this study on grammatically sound and well-written documents, we use Wikipedia articles of each relevant entity as a basis for candidate sentences. These are taken from the 2012 Wikipedia Wex dump. To obtain a base set for assessment, these sentences are processed by the ClausIE extraction system.

We ask assessors to imagine they were to write a knowledge article on the topic Q, on which they were to include information about the given entity e_i. Assessors are asked to mark passages that would be suitable support passages for the article by answering the following question:

AQ1) Explanation: Does the sentence explain the relevance of entity e_i?

This way we obtain candidate sentences for 95 query-entity pairs as input topics. We arrive at a total of 31,397 assessed sentences with 2,906 relevant support passages of entity relevance. Often, the relevant aspects of a relevant entity are not noteworthy enough to be described in the entity's article [6]. This leads to 20 query-entity pairs that don't contain any explanations of entity-relevance. These are excluded from this study, leaving 75 query-entity pairs and 22,731 support passage annotations of which 2,906 are marked as relevant according to AQ1.

Table 2: Performance of AQ1–5 as predictors for explanations and Pearson correlation ρ. $*$ significance over Qterm

	Relation	Rel rel	ClausIE	ClausIE rel	Qterm (*)	Name
Prec(*)	0.46 ±0.05	**0.52** ±0.05 *	0.45 ±0.05	0.49 ±0.05 *	0.38 ±0.04	0.33 ±0.05
Recall	0.28 ±0.03	0.21 ±0.03	0.20 ±0.02	0.14 ±0.02	**0.49** ±0.04	0.43 ±0.04
ρ	0.27	**0.52**	0.33	0.49	0.47	0.35
Count	1767 (8%)	935 (4%)	1172 (5%)	636 (3%)	4476 (20%)	6173 (27%)

Table 3: Results on ranking of sentences explaining entity relevance with LTR.

Method	MAP (*)	Hurt	Helped	Ablation	MAP
Full	**0.44** ±0.03	–	–		
Text	0.42* ±0.03	23	9	Full-TEXT	0.41 ±0.03
NLP	0.31* ±0.03	39	11	Full-NLP	0.43 ±0.04
DP	0.33* ±0.03	43	5	Full-DP	0.43 ±0.04
ClausIE	0.41* ±0.03	25	11	Full-ClausIE	0.43 ±0.03

In order to study characteristics of sentences in relation to AQ1, we further ask annotators to assess the following questions for each sentence s_{ik}, per query Q and entity e_i:

AQ2) Relation: Does the sentence mention any relationship involving e_i?
AQ3) Rel rel: Is this relationship relevant for the explanation?
AQ4) ClausIE: Does ClausIE extract a valid relationship from sentence?
AQ5) ClausIE rel: Is ClausIE's extraction relevant for the explanation?

We study these annotations in combination with two heuristics:
Qterm: Does the sentence include query terms (stopwords ignored)?
Name: Does the sentence include the entity's name?

5.2 Experiment 1: Relations and Relevance

By casting the result of every annotation question (Relation, Rel rel, ClausIE, ClausIE rel) as well as heuristics (Qterm, Name) as a random variable, we study both the Pearson correlation ρ of these predictors and the ground truth (Explanation / AQ1) as well as their predictive power as measured by set precision and recall in Table 2.

These demonstrate that good explanations are found in sentences that express a relevant relation of the entity (Rel rel / AQ3), reflected in the highest Pearson correlation of 0.52, as well as the highest precision of 0.52. Using Rel rel as a predictor is significantly[5] better in terms of precision than using the Qterm heuristic (which achieves precision of 0.38). However, the Qterm heuristic achieves a much higher recall of 0.49. This suggests that combining query terms and relation extractions is a worthwhile avenue for investigation.

Of course, this requires an automatic approach for distinguishing *relevant* from non-relevant relation expressions. On the pessimistic side, only half of all extracted relations are indeed relevant. On optimistic side, macro-avg precision drops only mildly from 0.52 for relevant relations (Rel rel / AQ3) to 0.46 for any relation (AQ2) and 0.45 for ClausIE extractions (AQ4). We speculate that an OpenIE relation extractor can also serve as a quality indicator for passages as it is sensitive towards well-formed sentences.

[3] data set available: www.cs.unh.edu/~dietz/appendix/openie4ir
[4] http://mschuhma.github.io/rewq/
[5] Paired-t-test with $\alpha = 5\%$.

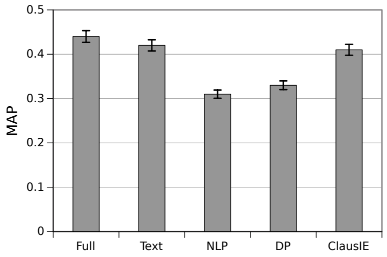

Figure 1: Results on ranking of sentences explaining entity relevance: full vs. subsets

5.3 Experiment 2: Evaluation through LTR

Next we demonstrate that features derived from ClausIE's extractions can be effectively used to train a learning-to-rank (LTR) method for ranking support passages as detailed in Section 4. The *Full* feature set, given in Table 1, is divided into four feature sets by category: Text, NLP, DP, and ClausIE. We compare our approach using all features (*Full*) versus each feature set individually. Statistically significant improvements of the *Full* using a paired t-test (α = 5%) are marked with *. We additionally perform an ablation study by removing one feature subset at a time (Full-*category*) from the *Full* feature set, to study redundancy in the feature space.

For learning to rank, approaches are evaluated with 5-fold cross-validation, where all rankings associated with the same query (but different entities) are assigned to the same fold. The ranking performance is measured in mean-average precision (MAP) with respect to the ground truth of a sentence explaining the relevance of the entity for the query (AQ1). Results are presented in Table 3 and Figure 1. Unjudged sentences are considered non-relevant. Results given in Table 3 show that the *Full* method outperforms all other methods significantly with a MAP of 0.36. Individually, the strongest feature subsets are *ClausIE* and *Text*, and the ablation study confirms that they provide complementary merit.

Despite issues due to precision-orientation of OpenIE systems (more details about this in the next section), we obtain significant improvements with respect to the recall-oriented evaluation metric MAP. This demonstrates that there is merit in further investigating high-level NLP extractions based on OpenIE. This is in contrast to other kinds of NLP extractions such as POS tags, NER tags, and dependency parse information which are significantly worse indicator for support passages.

5.4 Experiment 3: Open Issues

Many NLP-oriented systems are tuned for high precision at the expense of recall. While this is a desirable property in the context of knowledge base population, it may impose limitations for information retrieval tasks.

Among all sentences that express a relation, ClausIE is missing this relation in 32% of the cases. Additionally, only half of the sentences with relation expressions actually actually contain a relation that is relevant for the query-entity pair (confirming findings of Schuhmacher et al. [12]). Together this results in only 636 sentences with relevant ClausIE extractions (3%) of all 22731 annotated sentences. In contrast, our data set contains 2906 sentences (13%) with explanations of relevance.

While there are ClausIE extractions for 9951 sentences, only 1172 constitute a correct extraction. Comparing this to the 2906 true relevant sentences demonstrates that a perfect recall is not obtainable. Let us consider an optimistic thought experiment where all sentences with correct ClausIE extractions are relevant. An ideal ranking, which places all relevant sentences first, would obtain a MAP value of $\frac{1172}{2906} = 0.41$ (theoretical upper bound). This upper bound happens to coincide with the actual MAP achieved by the ClausIE feature set alone, MAP 0.41, cf. Table 3. We conclude that our approach obtains an optimal ranking under limitations imposed by the off-the-shelf OpenIE system. Improving coverage of OpenIE systems is likely to translate to immediate quality improvements for text-ranking tasks.

6 CONCLUSION

We study the utility of OpenIE technology ranking sentences by how well they explain the relevance of a given entity for a query. Based on manual assessments and evaluation through a learning-to-rank framework, the study demonstrates that significant improvements are achieved by combining relation features with query and entity matches. While we demonstrate the merit of an OpenIE extraction system, we also quantify losses through limitations of current OpenIE systems. we hope this study stimulates work on relation extraction systems that are designed of information retrieval tasks.

Acknowledgements

This research was performed as part of a Master's thesis at Mannheim University, Germany. We are grateful to Rainer Gemulla for providing access to the ClausIE system. This work was funded in part by a scholarship of the Eliteprogramm for Postdocs of the Baden-Württemberg Stiftung.

REFERENCES

[1] J. Aslam, F. Diaz, M. Ekstrand-Abueg, R. McCreadie, V. Pavlu, and T. Sakai. Trec 2014 temporal summarization track overview. Technical report, 2015.

[2] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction from the web. In *Proc. of IJCAI*, 2007.

[3] A. Berntson et al. Providing entity-specific content in response to a search query, Mar. 8 2012. US Patent App. 12/876,638.

[4] R. Blanco and H. Zaragoza. Finding support sentences for entities. In *Proc. of SIGIR*, 2010.

[5] L. Del Corro and R. Gemulla. Clausie: clause-based open information extraction. In *Proc. of WWW*, 2013.

[6] L. Dietz, A. Kotov, and E. Meij. Tutorial on utilizing knowledge graphs in text-centric information retrieval. In *Proc. of WSDM*, 2017.

[7] A. Fader, S. Soderland, and O. Etzioni. Identifying relations for open information extraction. In *Proc. of EMNLP*, 2011.

[8] C. Kedzie, K. McKeown, and F. Diaz. Predicting salient updates for disaster summarization. In *Prof. of ACL*, 2015.

[9] X. Liu and H. Fang. Latent entity space: a novel retrieval approach for entity-bearing queries. *Information Retrieval Journal*, 18(6):473–503, 2015.

[10] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. The Stanford CoreNLP natural language processing toolkit. In *Proc. of ACL*, 2014.

[11] B. Roth, T. Barth, G. Chrupa la, M. Gropp, and D. Klakow. Relationfactory: A fast, modular and effective system for knowledge base population. In *Proc. of EACL*, 2014.

[12] M. Schuhmacher, B. Roth, S. P. Ponzetto, and L. Dietz. Finding relevant relations in relevant documents. In *Proc. of ECIR*, 2016.

[13] M. Surdeanu, M. Ciaramita, and H. Zaragoza. Learning to rank answers to non-factoid questions from web collections. *Computational Linguistics*, 37(2):351–383, 2011.

[14] N. Voskarides, E. Meij, M. Tsagkias, M. de Rijke, and W. Weerkamp. Learning to explain entity relationships in knowledge graphs. In *Proc. of ACL*, 2015.

[15] L. Yao, A. Haghighi, S. Riedel, and A. McCallum. Structured relation discovery using generative models. In *Proc. of EMNLP*, 2011.