Processing Document Collections to Automatically Extract Linked Data: Semantic Storytelling Technologies for Smart Curation Workflows

Peter Bourgonje, Julian Moreno Schneider, Georg Rehm, Felix Sasaki
DFKI GmbH, Language Technology Lab
Alt-Moabit 91c,
10559 Berlin, Germany
{peter.bourgonje,julian.moreno_schneider,georg.rehm,
felix.sasaki}@dfki.de

Abstract

We develop a system that operates on a document collection and represents the contained information to enable the intuitive and efficient exploration of the collection. Using various NLP, IE and Semantic Web methods, we generate a semantic layer on top of the collection, from which we take the key concepts. We define templates for structured reorganisation and rearrange the information related to the key concepts to fit the respective template. The use case of the system is to support knowledge workers (journalists, editors, curators, etc.) in their task of processing large amounts of documents by summarising the information contained in these documents and suggesting potential story paths that the knowledge worker can then process further.

1 Introduction and Context

Journalists writing a story typically have access to large databases that contain information relevant to their topic. Due to the novelty requirement, they are under pressure to produce a story in a short amount of time. They have to provide relevant background, but also present information that is new and eye-opening. Curators who design showrooms or museum exhibitions often cannot afford to spend much time on getting familiar with a new domain, due to the large variety of unrelated domains they work in. Several other job profiles rely on extracting key concepts from a large document collection on a specific domain and understanding how they are related to one another. We refer to the group of people facing this challenge as *knowledge workers*. The common ground of their tasks is the *curation of digital information*. In our two-year project we collaborate with four SME partner companies that cover four different use cases and sectors (Rehm and Sasaki, 2015). We develop technologies that enable knowledge workers to delegate routine tasks to the machine so that they can concentrate on their core task, i. e., producing a story that is based on a specific genre or text type and that relies on facts contained in a document collection. Among the tools that we develop and integrate into the emerging platform are semantic storytelling, named entity recognition, entity linking, temporal analysis, machine translation, summarisation, classification and clustering. We currently focus on making available RESTful APIs to our SME partners that provide basic functionalities that can be integrated into their own in-house systems. In addition, we work on implementing a system for semantic storytelling. This system will process large document collections, extract entities and relations between them, extract temporal information and events in order to automatically produce a hypertext view of the collection to enable knowledge workers to familiarise themselves with the document collection in a fast and efficient way. We also experiment with automatically generating story paths through this hypertext cluster that can then be used as the foundation of a new piece of content. The focus of this paper is on the approach we use to fill templates that assist the user in the generation of a story and on selecting possible topics for a new story.

2 Related Work

The emerging platform we develop connects all RESTful APIs that perform the individual analyses by using the FREME framework (Sasaki et al., 2015) throughout. A system with a similar setup is described in (Lewis et al., 2014), but unlike our platform, this is mainly targeted at the localisation industry and it deploys a different approach to representing curated data. Other systems aimed at collecting and processing semantically enriched content are developed in the context

of the NewsReader project [1], specifically targeted at the news domain and the SUMMA project [2], focusing on broad coverage of languages through the use of machine translation. Our approach towards language generation consists of filling templates to create building blocks for a story (which are not grammatical sentences), and it deviates from a.o. (Cimiano et al., 2013), (Galanis and Androutsopoulos, 2007), (Bontcheva and Wilks, 2004) in that it does not focus on one specific domain and it includes collecting the information that goes into the ontology. Our approach is based on extracting relations between entities. Because of our focus on domain adaptability, we plan to avoid the labour-intensive selection of seed patterns that comes with (semi-)supervised approaches as described in (Xu et al., 2013), (Brin, 1998), (Agichtein and Gravano, 2000) and (Etzioni et al., 2005). We use an unsupervised approach instead with an open set of relation types. A similar system is described in (Yates et al., 2007). For ease of integration reasons we implement an in-house approach to relation extract, as described in 4.

3 Linked Data Generation

We produce a semantic layer over the document collection consisting of a set of annotations for every document, represented in NIF[3]. Frequent named entities are interpreted as key concepts. An essential step in producing the semantic layer is therefore the spotting and linking of named entities. One of our APIs performs entity spotting based on pre-trained sample models (Nothman et al., 2012) and linking based on DBPedia Spotlight [4]. Because the users of our tools work on a variety of specific domains, the platform provides the possibility to train a model or to plug in a domain-specific ontology and upload a key-value structured dictionary. The NER module is based on the corresponding Apache OpenNLP module (Apache Software Foundation, 2016). The NER API generates a NIF representation of the input in which every recognised entity is annotated with the corresponding URI in DBPedia. Specialised SPARQL queries are used to retrieve type-specific related information (like birth and death dates and places for persons, geo-coordinates for locations, etc.). This information is added to the semantic layer and is used to fill particular slots in our story templates.

4 Semantic Storytelling

For the task of storytelling we use a template-filling approach. The user selects one or more concepts and one or more templates that we attempt to fill using the semantic layer. We have defined a *biography* template and an *event* template. The biography template contains the following slots: full name, pseudonym, date of birth, place of birth, date of death, place of death, mother, father, siblings, spouse, children, occupation, key persons and key locations. The event template contains slots for date(s), key persons and key locations. To collect the information needed to fill the individual slots in the templates, the results of the type-specific SPARQL queries are used. Information in the ontology is typically of higher quality than information extracted using a relation extraction component. However, because the user often works in domains for which no ontology is available, relation extraction is used to search for the missing information in the document collection. In addition to filling slots in templates, the output of relation extraction between entities allows the user to learn about relations between key concepts that are not directly related to any slots in the templates. This allows the user to get a better understanding of the document collection, but can also be the basis for selecting and populating a new event template, hence generating an angle for a new story. Relations are extracted in the following way: A document collection is assumed to be relatively homogeneous from a content perspective (file types and information type (running text or metadata) may vary, but the information in a collection is assumed to be about one domain). The goal is to aggregate information and extract relevant relations regardless of which document these originate from. The assumption is that in the NER procedure all relevant concepts have been annotated and are present in the semantic layer. For every sentence containing two or more entities, a dependency tree is generated using the Stanford CoreNLP DependencyParser (Manning

[1] http://www.newsreader-project.eu/

[2] http://www.summa-project.eu/

[3] See http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core/nif-core.html

[4] https://github.com/dkt-projekt/e-NLP contains code and documentation

et al., 2014). The entities are located in the tree, and a relation between entity A and entity B is established if A has a subject-type dependency relation to a verb node in the dependency graph and B has an object-type dependency relation to this same verb node in the graph. The value of this verb node (e.g. the token) is taken to express the relation. For collecting missing information related to specific slots in the template we plan to include a dedicated relation extraction system and train it with a number of seeds that correspond directly to the slots in the template we want to fill. Because of the limited number of relation types that are needed for this, a (semi-)supervised, seed-based approach is likely to produce useful results. For selecting new angles for a story (in the form of events as the basis for a new template), we want to capture a larger set of relation types. Since the domain typically determines the predominant relation types present in a collection, this calls for a more open approach that is not limited by the relation types covered by the seeds. To give an idea of the current stage of our relation extraction component, a demo-interface based on the Viking corpus is available at: `http://dev.digitale-kuratierung.de/ds/selectstory.php`.

5 Evaluation

To assess the suitability of the platform for the tasks of template filling and interactive collection exploration, we have asked four humans to perform these tasks and in the process evaluate the platform. For evaluation, three collections were used, two of them real-world use-cases provided by the SME partners of the project: (i) a collection of letters sent by the architect Erich Mendelsohn and his wife [5] and (ii) a private document collection on Vikings. The third collection is the WikiWars corpus [6].

The subjects were asked to check how many slots in the templates could be filled with the relations extracted. For the biography template some relevant relations were found in the Viking corpus: *Edward, marry, Edith*, which can fill the spouse slot and *William, raise, Malcolm*, which could fill the children slot (though it of course does not imply that Malcolm really is a child of William). Because we have not created a gold standard from these corpora, measuring recall is problematic. As a result, it is not clear whether the approach failed to extract the relations that could lead to populating templates, or whether this information was not present in the corpus. It is clear however that the real-world scenario would primarily rely on getting information from the ontology. An observation that was made by all test subjects was that the quality of the relations extracted from the WikiWars and the Viking corpus was much higher than that of relations extracted from the Mendelsohn corpus. This is probably due to the fact that the first two corpora are meant to be descriptive and clear records of historical events, whereas the Mendelsohn corpus consists of private letters that were probably not intended to be descriptive for the general public. In addition, the Mendelsohn letters contained many cases of *you* and *I*, which were not recognized by the NER component (because resolution to a URI is not in all cases straightforward), hence these sentences were not considered. With regard to the task of exploring the document collection and selecting possible events to base a new story on, the subjects reported that several useful relations were extracted, again with the exception of the Mendelsohn corpus. Examples are *[William [[extort, Harold], [leave, Aldred]]]* from the Viking corpus and *[Rome, annex, [Sardinia, Corsica]]* and *[Tripartite pact, unite, [Italy, Germany, Japan]]* from the WikiWars corpus. These pieces of information also display the advantage of a non-seed-based approach. To capture the above relations, we would need to have trained for relations of the extort-, leave-, annex- and unite-type. The point however is that upfront the potentially interesting relations that need to be extracted are of unknown types.

A drawback of the current approach is the fact that only binary relations (a predicate with two arguments) are extracted. Ditransitive verbs are not fully captured. Instead, we end up with relations like *[Edward, give, William]*, where an important piece of the expressed relation is missing. Another drawback is the requirement that the two entities are connected through the same verb node in the dependency graph. We have no concrete figures on recall, but the number of extracted relations were relatively low for all corpora. A third observed issue were errors introduced by the dependency parser. Relations like *[United States, mem-*

[5] `http://ema.smb.museum/en/home/`

[6] `http://timexportal.wikidot.com/forum/t-275092/wikiwars-is-available-for-download`

ber, Allies] and *[Poland, better, Russian Empire]* were extracted, where apparently *member* and *better* were tagged as a verb, hence considered by our analysis.

6 Conclusion

We present a platform that generates a semantic layer over a document collection and applies relation extraction to fill templates that serve as building blocks for the generation of new stories. This information is extracted, analysed, in some cases rearranged and presented to the user in the form of (i) filled out templates or (ii) through an interactive tree-based exploratory view. Test users pointed out its usefulness to exploring the contents of a collection in a fast and intuitive way and its shortcomings with regard to non-binary relations, low recall and lack of parsing precision. The most important suggestions for future work are looking into means of extracting indirect objects (or generally relevant additional information for specific relations, such as location or time) and defining paths through the dependency graphs rather than requiring direct connections in the graph. In addition we plan to plug in a verb ontology to be able to group the different types of relations we find.

References

Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the Fifth ACM Conference on Digital Libraries*, DL '00, pages 85–94, New York, NY, USA. ACM.

Apache Software Foundation. 2016. Apache OpenNLP, `http://opennlp.apache.org`.

Kalina Bontcheva and Yorick Wilks. 2004. Automatic report generation from ontologies: The MIAKT approach. In Farid Meziane and Elisabeth Métais, editors, *Natural Language Processing and Information Systems, 9th International Conference on Applications of Natural Languages to Information Systems, NLDB 2004, Salford, UK, June 23-25, 2004, Proceedings*, volume 3136 of *Lecture Notes in Computer Science*, pages 324–335. Springer.

Sergey Brin. 1998. Extracting patterns and relations from the world wide web. In *In WebDB Workshop at 6th International Conference on Extending Database Technology, EDBT98*, pages 172–183.

Philipp Cimiano, Janna Lüker, David Nagel, and Christina Unger. 2013. Exploiting ontology lexica for generating natural language texts from rdf data. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 10–19, Sofia, Bulgaria, August. Association for Computational Linguistics.

Oren Etzioni, Michael Cafarella, Doug Downey, Ana Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. 2005. Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, 165(1):91–134, 6.

D. Galanis and I. Androutsopoulos. 2007. Generating Multilingual Descriptions from Linguistically Annotated OWL Ontologies: the NaturalOWL System. In *Proceedings of the 11th European Workshop on Natural Language Generation (ENLG 2007)*, pages 143–146, Schloss Dagstuhl, Germany.

David Lewis, Asunción Gómez-Pérez, Sebastian Hellman, and Felix Sasaki. 2014. The role of linked data for content annotation and translation. In *Proceedings of the 2014 European Data Forum*, EDF '14.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Joel Nothman, Nicky Ringland, Will Radford, Tara Murphy, and James R. Curran. 2012. Learning multilingual named entity recognition from Wikipedia. *Artificial Intelligence*, 194:151–175.

Georg Rehm and Felix Sasaki. 2015. Digitale Kuratierungstechnologien – Verfahren für die effiziente Verarbeitung, Erstellung und Verteilung qualitativ hochwertiger Medieninhalte. In *Proceedings of the 2015 International Conference of the German Society for Computational Linguistics and Language Technology*, GSCL '15, pages 138–139.

Felix Sasaki, T. Gornostay, M. Dojchinovski, M. Osella, E. Mannens, G. Stoitsis, Philip Richie, T. Declerck, and Kevin Koidl. 2015. Introducing freme: Deploying linguistic linked data. In *Proceedings of the 4th Workshop of the Multilingual Semantic Web*, MSW '15.

Feiyu Xu, Hans Uszkoreit, Hong Li, Peter Adolphs, and Xiwen Cheng, 2013. *Domain Adaptive Relation Extraction for Semantic Web*, chapter X. Springer.

Alexander Yates, Michael Cafarella, Michele Banko, Oren Etzioni, Matthew Broadhead, and Stephen Soderland. 2007. Textrunner: Open information extraction on the web. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, NAACL-Demonstrations '07, pages 25–26, Stroudsburg, PA, USA. Association for Computational Linguistics.