StoryBlink
a Semantic Web Approach for Linking Stories

Ben De Meester, Tom De Nies, Laurens De Vocht,
Ruben Verborgh, Erik Mannens, and Rik Van de Walle

Ghent University – iMinds – Multimedia Lab, Belgium
{firstname.lastname}@ugent.be

Abstract. Finding relevant content automatically is not straightforward due to the unstructured nature of large text corpora. Moreover, traditional techniques to extract structured information out of these corpora are mostly very fine-grained, which deteriorates the needed high-level overview to compare publications. Also, publishing this information as Linked Data can provide for very important context information. This demo paper describes "StoryBlink", a Web application that enables the discovery of stories through linked books. By finding paths between stories that are represented by a compact set of semantic concepts, it provides the user with relevant stories, based on previously selected publications. By also returning the common concepts between these stories, it gives the user a quick insight into how certain stories are connected. As such, StoryBlink enables an automatic content-based discovery journey between linked stories.

Keywords: Linked Data, NLP, Path Finding

1 Introduction

The continuous increase of publications – albeit printed or digital, journals or novels – needs better and better systems that find relevant publications for an end user[1]. This is usually achieved by comparing publications, and discovering its similarities. In this paper, the two most important functional requirements for this comparison are having an automatic system (to cope with the huge amount of published content), and comparing these publications regarding their most important features. In the case of stories, one of the most important features is its content. For example, recommending a publication solely because it is written by the same author will not always give a desired result, as this recommended publication could handle completely different content.

StoryBlink[2] automatically discovers relevant publications, and provides the user with these relevant publications and the semantic links between them. Each publication is represented as a set of concepts that are prevalent in the publication. Based on these concepts, strong and weak connections can be found between publications. The user can interpret these connections to assess the relevancy of a publication.

[1] Relevancy is from hereon defined as containing similar content.
[2] http://uvdt.test.iminds.be/storyblink/

2 Related work

To automatically find relevant publications, it is necessary to extract structured data (e.g., XML or RDF) from plain text. Natural Language Processing (NLP) is concerned with all interactions between computers and natural languages. NLP is a very broad field, of which two tasks are applicable to create a machine-interpretable representation of a large piece of text (i.e., the story of the publication), namely (i) understanding natural language sentences in terms of, e.g., mentioned concepts and relations, and (ii) summarizing natural language into a smaller yet just as informative text.

Mechanisms to understand natural language sentences – such as Named Entity Recognition and Disambiguation (NER and NED) – provide a machine-interpretable representation of mentioned concepts [1]. However, this research domain is currently mostly targeted at small pieces of text (e.g., paragraphs and tweets), and the resulting data is too detailed, for every sentence, that it becomes no longer practical to work with the entire resulting data set. A more high-level representation of a piece of text is, to the best of our knowledge, not possible using current techniques.

Summarization of text is the research domain of representing a large corpus into a smaller yet just as relevant text [4]. Although semantic solutions are in use (i.e., take the semantic meaning of words into account to improve the summarization result), NLP summarization methods are targeted at human use, i.e., the resulting text is a human-understandable piece of text. To the best of our knowledge, no summarization techniques have been developed that return a short representation of a large text corpus in a machine-interpretable format.

3 Technical overview

By executing NER and NED engines on the text of digital publications, and filtering the resulting tags, we can return the most important concepts that a story comprises in a machine-interpretable (i.e., semantic) format. These tags can be used to link publications based on their content. Moreover, as this representation can be published as Linked Data, we can use the context of these concepts to better assess why certain stories are linked to each other, e.g., knowing that two stories mention the French city Paris is more valuable than knowing that two stories mention the word Paris (which could mean the city in one story and the Hollywood personality in the other). The Web application StoryBlink shows how this can be used as a discovery mechanism for related publications.

We choose to detect the concepts of the full text of the publication, and filter out the most important concepts, instead of detecting the concepts of a summary. Otherwise, the quality of the set of representative tags would be dependent of the quality of the summary. Also, this second approach would not allow to link a detected concept with its original context, whilst the first approach does.

We use DBpedia Spotlight [3] as NER and NED engine. Other disambiguation engines exist, but we choose DBpedia Spotlight, as it is open-source, and automatically connects the detected concepts to their URI on `http://dbpedia.org`. DBpedia Spotlight – being only able to link to DBpedia concepts – is biased towards famous people and places, and is not capable to disambiguate all fictional characters. This is however

no blocking factor, as StoryBlink's aim is to connect books based on their common concepts, and fictional characters are rarely common concepts between different stories. The resulting URIs are then exported to RDF to describe a book using a set of DBpedia URIs as representative tags, e.g., `:book :mentions :Paris`.

Since extracting all detected concepts in a publication inevitably introduces noise, we use a filter. This filter keeps the most frequently detected concepts. The amount of detections per concept are counted per publication, and only the most mentioned concepts that account for 50% of the total amount of detections are included in the final set of representative tags. This filter step reduces the total amount of triples from 22 132 to 1 091, whilst still finding 94.06% of all potential paths. After processing each publication, these individual results are then published together in one Triple Pattern Fragments endpoint[3]. Triple Pattern Fragments is used as it allows hosting and querying Linked Data in an affordable and reliable way [5], and thus allows for the creation of Linked Data applications on top of live endpoints. The current endpoint houses the analysis results of 20 classic works, as found on Project Gutenberg[4].

We can then use this endpoint to find connections between relevant publications. Using the Everything is Connected engine [2], it is possible to find relevant paths between two points in a graph. In this paper, the Everything is Connected engine finds paths between two selected books by comparing their sets of DBpedia links, and possibly using intermediate books as linking point. These paths are weighted depending on the amount of tags two books have in common, and depending on the amount of hops between them for a given path. Thus, direct links are found between books using a common set of representative tags, or indirect links are found using related books as intermediate nodes.

4 StoryBlink

StoryBlink is available at `http://uvdt.test.iminds.be/storyblink/`. After selecting two publications, the user can click "StoryBlink!" to let the Everything is Connected engine find relevant paths between the two chosen publications (as illustrated in Figure 1). When measuring the calculation time for finding the paths between 190 different pairs of publications, we can conclude that the average calculation time is 5.28s. Taken into account that all calculations are done on the fly, using a public Triple Pattern Fragments endpoint, this calculation time is reasonable.

The resulting paths – and in-between publications – are then visualized. All publications (including the two starting publications) are connected according to the found paths, and the width of the links denote the link strength. As such, StoryBlink provides an overview of the links and their strengths between all relevant publications. By clicking on a link between publications, StoryBlink will query the data endpoint to ask for the commonalities between these two publications. The resulting concepts give the user an indication on why these two publications are connected, and thus help the user to assess whether the linked publication is relevant for him or her.

[3] `http://uvdt.test.iminds.be/storyblinkdata/books`

[4] `http://www.gutenberg.org/`

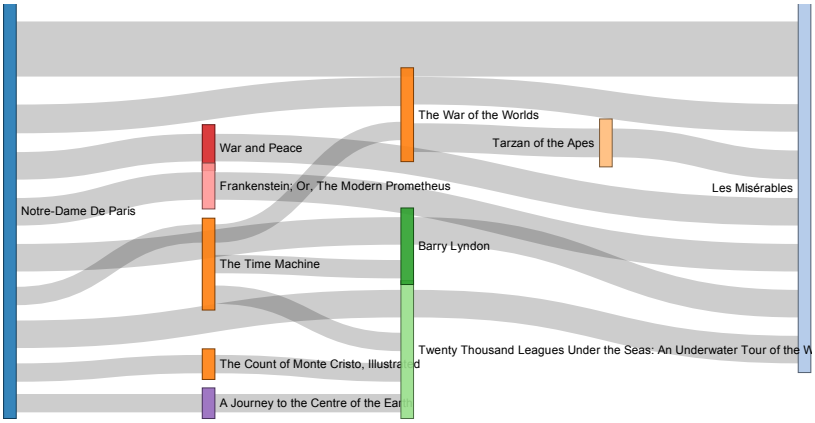

Fig. 1. StoryBlink shows relevant links between classic works in a diagram where different stories are linked to each other, and where the weight of the paths is indicated by the width of the links.

5 Conclusion

Using StoryBlink, a user can discover links between classic works. Adding extra publications to StoryBlink requires minimal effort as the semantic summaries are created automatically. StoryBlink thus enables users to discover publications from the backlog of a publisher, without biasing the results towards popular works, as is usually the case with, e.g., social recommendation systems. Future work includes improving the performance, evaluating the filtering method (e.g., comparing it to tf-idf), and making use of named entity recommendation systems to find paths between publications that have related concepts in common instead of exactly the same concepts. Other knowledge bases could be used to prevent DBpedia's bias towards famous people and places.

References

1. Cucerzan, S.: Large-scale Named Entity Disambiguation based on Wikipedia data. In: EMNLP-CoNLL. vol. 7, pp. 708–716 (2007)
2. De Vocht, L., Coppens, S., Verborgh, R., Vander Sande, M., Mannens, E., Van de Walle, R.: Discovering meaningful connections between resources in the Web of Data. In: Bizer, C., Heath, T., Berners-Lee, T., Hausenblas, M., Auer, S. (eds.) Linked Data on the Web (LDOW). pp. 1–8. CEUR, Rio De Janeiro, Brazil (May 2013)
3. Mendes, P.N., Jakob, M., García-Silva, A., Bizer, C.: DBpedia Spotlight: Shedding light on the Web of documents. In: Proceedings of the 7th International Conference on Semantic Systems. pp. 1–8. ACM (2011)
4. Nenkova, A., McKeown, K.: A survey of text summarization techniques. In: Aggarwal, C.C., Zhai, C. (eds.) Mining Text Data, pp. 43–76. Springer US (January 2012)
5. Verborgh, R., Hartig, O., De Meester, B., Haesendonck, G., De Vocht, L., Vander Sande, M., Cyganiak, R., Colpaert, P., Mannens, E., Van de Walle, R.: Querying datasets on the Web with high availability. In: International Semantic Web Conference 2014. pp. 180–196. Springer (2014)