

Masters Thesis Exposé in Computer Science

Extending Semantic Hypergraphs by Neural Embedding-based Semantic Similarity for Pattern Matching

Max Reinhard

Matrikelnummer: 359417

June 23, 2023

Supervised by Prof. Dr. Manfred Hauswirth
Additional guidance by Prof. Dr. Camille Roth* and Dr. Thilo Ernst[†]

*Centre Marc Bloch (An-Institut der Humboldt-Universität zu Berlin)

[†]Fraunhofer-Institut für offene Kommunikationssysteme

1 Introduction

A significant part of the social world is nowadays being represented by digitally manifested text. Examples for this range from instant messaging, social media and any form of collective web activity to encyclopaedic websites, digitized libraries and government intelligence. The amount and richness of available social text makes it a valuable data source for social science research and simultaneously creating an interest in automatic systems to analyze these texts on a large scale (Evans and Aceves 2016). Such research can be understood as part of the domain of *Computational Social Science* (CSS) (Lazer et al. 2009).

Systems based on techniques from the field of *Natural Language Processing* (NLP), as well as the interlinked fields of *Information Retrieval* (IR) and *Information Extraction* (IE), have demonstrated great success in a variety of task related to text analysis. This success is large attributed to the advancements of applying of *Machine Learning* (ML) and especially *Deep Learning* (DL) methods to text (Hirschberg and Manning 2015) (Qiu et al. 2020). While being very effective at predicting or decision making, ML- and specifically DL-based systems generally do not deliver an explanation for their judgement and can mostly be viewed as "black box models" that are not transparent in their decision making process (Rudin 2019). Conversely this transparency and explainability is of high interest in CSS applications such as predicting political opinion based on social media activity (Wilkerson and Casas 2017).

The *Semantic Hypergraph* (SH) (Menezes and Roth 2021) is a framework for representing and analyzing Natural Language (NL). *NL* sentences can be modelled as an ordered, recursive hypergraph which can be represented in a formal language. The framework allows to specify semantic patterns in this formal language which can be matched against an existing SH. It aims to provide an *open* and *adaptive* system to analyse text corpora, especially in the domain of CSS. The framework is *open* in the sense that its representation formalism is inspectable and intelligible by humans and that the pattern matching follows explicit rules. The framework is *adaptive* in the sense that the parsing is built from adaptive, ML-based subsystems and therefore allows for an error-tolerant parsing from *NL* to *SH* in regards to grammatical and syntactical correctness.

2 Problem Statement

In the SH framework, text is represented in the form of *hyperedges* (in the following also referred to as *edges* only). These edges are either atomic or they consist of edges themselves, which essentially accounts for the recursive character of the SH. Each edge has a specific *type* from a set of eight different types of which the most trivial two types are probably *concept* (C) and *predicate* (P).

Users of the SH framework (e.g. CSS researchers) can define patterns in the SH formalism to match against a text corpus (e.g. news articles) that has previously been parsed as an SH. These patterns may among other things specify the structure of the edges that should match it as well as their type (and the types of possible sub-edges). Additionally the actual words that should match need to be specified i.e. the content to match against, if the structure of an edge matches the pattern. There are additional operators in the pattern language such as the wildcard operator ***, which can be used e.g. to match every edge of a specific type and therefore discard content.

To better illustrate the problem Hyperedge 2 and Hyperedge 1 demonstrate how NL sentences are parsed to SH based on this simplified introduction the the SH representation.

(likes/P ann/C apples/C)

Hyperedge 1: SH representation for the sentence "Ann likes apples"

(likes/P ann/C bananas/C)

Hyperedge 2: SH representation for the sentence "Ann likes bananas"

Hyperedge 1 and Hyperedge 2 both follow the same structure, but differ in the content of the last sub-edge. Both edges are hence matched by Pattern 1, which does not specify content for this sub-edge. The SH pattern language also allows to define a pattern that matches both Hyperedge 1 and Hyperedge 2 via a list of words as in Pattern 2. However is not possible define a pattern that matches based on some form of *Semantic Similarity* (SS) (Harispe et al. 2015) regarding content. Referring to the example above this means e.g. it is not possible to match an abstract semantic concept such as a *fruit*. Using the SH framework to retrieve every sentences that declares that "Ann likes some kind of fruit", would require to provide a comprehensive list of every fruit.

(likes/P Ann/C */C)

Pattern 1: "Ann-likes-something" pattern

(likes/P ann/C [apples, bananas]/C)

Pattern 2: "Ann likes apples or bananas" pattern

Word semantics generally depend on textual context and hence does the SS between words (Harispe et al. 2015, Section 2.2.3). Incorporating contextuality when extending the SH pattern matching process by SS therefore poses a central challenge. Context-dependent SS would allows to specify matching edge content beyond isolated word semantics, although is may not always be desirable or necessary as in the example above. Nevertheless one could modify this example so that it is desired to match sentences which state that "Ann likes *fruit*, because it is sour", while not being interested in the structure of the second clause. Providing a reference context by sentences like "Ann likes limes, since they are so sour" and "Ann likes lemons, because of their sourness", would possibly allow to capture the concept of *sour fruit* implicitly.

The necessity to explicitly list every word that should match the content of an edge restricts the generalisation capabilities of the SH frameworks pattern matching and introduces a bias via the selection of those very words. Utilizing some form of SS regarding to edge content for the matching step would allow users to define more generalising patterns, though analogously it would introduce a bias from the system utilized for SS computation.

There exists a great variety of approaches for determining the SS of texts of which recently ML-based and especially DL-based approaches have been outperforming most other approaches (Chandrasekaran and Mago 2021). They generally rely on computing a vector space representation or embedding of texts which can then be used to calculate their similarity and will therefore be referred to as *Neural Embedding-based* (NE) in the following.

As illustrated earlier, these approaches principally do not provide the explicitness that is inherent to the pattern matching process of the SH framework. In the sense of the

not sure whether to include this sentence

not sure whether to include this sentence

adaptive-open classification described above an integration of NESS would mean a shift from openness to adaptivity in this regard.

3 Research Questions

R Can neural embedding-based semantic similarity regarding edge content be integrated into the pattern matching of the Semantic Hypergraph framework to allow for more generalising patterns while providing control over the adaptiveness and therefore loss of explicitness in the matching process?

R.1 What neural embeddings model would be the most suitable for accurately assessing semantic similarity within the Semantic Hypergraph pattern matching process while effectively addressing the challenges posed by contextuality?

R.2 To what extent does incorporating neural embedding-based semantic similarity improve the generalization (recall) performance when matching a pattern against a set of known desired matching results?

R.3 How can adaptiveness and explicitness of the matching process be effectively and transparently balanced and controlled?

4 Proposed Solution

The solution proposed by this thesis to answer the above mentioned research questions is to conceptualize and implement a proof-of-concept and subject it to a suitable evaluation.

The implementation should fulfill the following criteria:

- Efficiently integrate NESS into the SH frameworks pattern matching process while maintaining its original functionality
- Allow for different NESS models to be used to compare their respective pattern matching results and enable integration of future developments in the field of NESS
- Enable the user of the SH framework to control the adaptiveness of the pattern matching through e.g. a parameter

Additionally the evaluation should fulfill the following:

- Show the difference in matching results between adaptive and standard pattern matching, both qualitatively and quantitatively.
- Show the difference in matching results for different NESS models, both qualitatively and quantitatively.
- Show the difference in matching results for the adaptiveness control system, both qualitatively and quantitatively.

add the notion of trying to capture abstract semantic concepts with semantic similarity to the research questions? this would then obviously also needed to be included earlier

add something about contextuality? e.g. how to provide context?

5 Approach

In order to implement the suggested solution, the following steps will be undertaken:

1. **Research** A study of the different types of NESS models will be carried out, paying particular attention to how they represent context. Those offering state-of-the-art performance and are best suited for integration into the SH pattern matching will be identified.
2. **Conceptualization** The most opportune point for NESS integration within the SH framework's matching process will be located. The parts of the SH framework that need modification for NESS integration, as well as the missing subsystems for employing NESS in the SH pattern matching process will be determined. This will take into account the requirements of the NESS models selected in the previous step.
3. **Implementation** Necessary subsystems will be added to the SH framework to enable NESS usage in the SH pattern matching. Additionally, the relevant parts of the SH framework will be modified.
4. **Evaluation** An evaluation framework will be constructed to perform the evaluations outlined earlier.

does this section need to be more specific?

6 Provisional Outline

- 1 **Introduction**
- 2 **Fundamentals and Related Work**
 - 2.1 Semantic Hypergraph
 - 2.2 Semantic Similarity
 - 2.3 Embedding-based Similarity
- 3 **Problem Statement**
 - 3.1 Research Questions
- 4 **Solution Approach**
 - 4.1 Integration into the Pattern Matching Process
 - 4.2 Fixed Word Embedding-based Matching
 - 4.3 Contextual Embedding-based Matching
 - 4.4 Similarity Threshold Control
- 5 **Implementation**
 - 5.1 Relevant external Software Libraries used
 - 5.2 Modules newly added to the SH Framework
 - 5.3 Modifications of the SH Pattern Matching
 - 5.4 Modifications to the Hypergraph database

6 Results and Evaluation

6.1 Case Study: Conflicts

7 Conclusion

8 Future Work

8.1 Implementation Improvements

8.2 Further Evaluations

7 Timeline

References

- Chandrasekaran, Dhivya and Vijay Mago (Feb. 18, 2021). “Evolution of Semantic Similarity—A Survey”. In: *ACM Computing Surveys* 54.2, 41:1–41:37. ISSN: 0360-0300. DOI: 10.1145/3440755. URL: <https://dl.acm.org/doi/10.1145/3440755> (visited on 06/17/2023).
- Evans, James A. and Pedro Aceves (July 1, 2016). *Machine Translation: Mining Text for Social Theory*. DOI: 10.1146/annurev-soc-081715-074206. URL: <https://papers.ssrn.com/abstract=2822747> (visited on 06/15/2023). preprint.
- Harispe, Sébastien et al. (2015). *Semantic Similarity from Natural Language and Ontology Analysis*. DOI: 10.2200/S00639ED1V01Y201504HLT027. arXiv: 1704.05295 [cs]. URL: <http://arxiv.org/abs/1704.05295> (visited on 06/19/2023).
- Hirschberg, Julia and Christopher D. Manning (July 17, 2015). “Advances in Natural Language Processing”. In: *Science* 349.6245, pp. 261–266. DOI: 10.1126/science.aaa8685. URL: <https://www.science.org/doi/abs/10.1126/science.aaa8685> (visited on 06/15/2023).
- Lazer, David et al. (Feb. 6, 2009). “Computational Social Science”. In: *Science* 323.5915, pp. 721–723. DOI: 10.1126/science.1167742. URL: <https://www.science.org/doi/full/10.1126/science.1167742> (visited on 06/15/2023).
- Menezes, Telmo and Camille Roth (Feb. 18, 2021). *Semantic Hypergraphs*. DOI: 10.48550/arXiv.1908.10784. arXiv: 1908.10784 [cs]. URL: <http://arxiv.org/abs/1908.10784> (visited on 07/19/2022). preprint.
- Qiu, XiPeng et al. (Oct. 1, 2020). “Pre-Trained Models for Natural Language Processing: A Survey”. In: *Science China Technological Sciences* 63.10, pp. 1872–1897. ISSN: 1869-1900. DOI: 10.1007/s11431-020-1647-3. URL: <https://doi.org/10.1007/s11431-020-1647-3> (visited on 06/15/2023).
- Rudin, Cynthia (May 2019). “Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead”. In: *Nature Machine Intelligence* 1.5 (5), pp. 206–215. ISSN: 2522-5839. DOI: 10.1038/s42256-019-0048-x. URL: <https://www.nature.com/articles/s42256-019-0048-x> (visited on 06/17/2023).
- Wilkerson, John and Andreu Casas (May 1, 2017). *Large-Scale Computerized Text Analysis in Political Science: Opportunities and Challenges*. DOI: 10.1146/annurev-polisci-052615-025542. URL: <https://papers.ssrn.com/abstract=2968080> (visited on 06/15/2023). preprint.