Technische Universität Berlin
Fakultät IV: Elektrotechnik und Informatik
Institut für Telekommunikationssysteme
Fachgebiet Verteilte offene Systeme

Masters Thesis in Computer Science

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

Max Reinhard
Matrikelnummer: 359417

January 11, 2024

Supervised by Prof. Dr. Manfred Hauswirth
Additional guidance by Prof. Dr. Camille Roth*
and Dipl.-Math. Thilo Ernst†

*Centre Marc Bloch (An-Institut der Humboldt-Universität zu Berlin)
†Fraunhofer-Institut für offene Kommunikationssysteme

Abstract Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Contents

1 Introduction

- Context: The big problem
- Problem statement: The small problem
- Methodology / Strategy
- Structure

Notes:

- Huge amounts of text, which can provide insight about stuff
- Automatic tools can provide assistance for humans to process all the text
- This generally means filtering the original text corpus or otherwise reducing amount of information the information that has to be processed by humans
- Filtering introduces a bias
- Especially for scientific purposes it is relevant to mitigate bias or at least understand what bias has been introduced (to make it transparent)
- Semantic Hypergraphs can be a valuable tool for that because...

Human life in times of widespread use of the internet and smartphones is most certainly more than ever interspersed with text-based communication...

A Semantic Hypergraphd (**menezes_semantic_2021**) is a form of representation for Natural Language (NL) and therefore knowledge. *NL* sentences can be modelled as a 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*.

The aim of the SH framework is to provide a *open* and *adaptive* framework to analyse text corpora, especially in the domain of computational social science (CSS) (**lazer2009computational**). The framework is *open* in the sense that it's 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 based on adaptive subsystems (ML-based) and therefore allows for an error-tolerant parsing from *NL* to *SH* in regards to grammatical and syntactical correctness (???).

2 Fundamentals and Related Work

2.1 Semantic Hypergraph

2.1.1 Structure

Edge

Edge Content

Root edge / Sequencen (i use the term "sequence root edge")

2.1.2 Syntax

wildcard operator

Variables

Square bracket notation

Differences between the formal notation and the notation sued in the implementation (or should this be contained in the implementation chapter?)

2.2 Semantic Similarity

2.2.1 Different Similarity Measures

String Similarity

Lievenstein distance, etc..

Lexical Similarity

tf-idf, etc.?

2.2.2 Types of Semantic Similarity

Lexical Databases

WordNet and alike (not the scope of this work)

2.3 Embedding-based Similarity

2.3.1 Embedding Types

Fixed Word Embeddings

Contextual Ebeddings

2.3.2 Distance Measures

Mean reference vector vs. pairwise distance

3 Problem Statement

CSS researches may typically be interested in retrieving statements of specific kind from a text corpus, such as expressions of sentiment of an actor towards some entity or expressions of conflicts between different actors. One approach for performing the retrieval would be to use a system which allows to specify some form of pattern which abstractly represents the statements they are trying to capture. This requires the definition of some form of formal pattern language[1] and possibly the prior transformation of the text corpus into some form of structured format to match against. Another approach is to use a system, which accepts example statements concretely representing the statements that are desired to be retrieved. Those systems may require a large number of positive and negative examples to be able to perform the retrieval. The two types of retrieval systems described here are in tendency situated in the realms of symbolic IR/IE and probabilistic ML/DL respectively.

The SH framework is more situated in the former symbolic realm. In SH 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. a collection of 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 atomic edge 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

[1] The *Google Search* query language can be seen as a simple example of such a pattern language, albeit with a different use case focus: `https://support.google.com/websearch/answer/2466433?hl=en`

is not possible define a pattern that matches based on some form of *Semantic Relatedness* (SR) or *Semantic Similarity* (SS) (Harispe et al. 2015) regarding content. Referring to the example above this means using the SH framework it is not directly possible to to retrieve every sentences that declares that "Ann likes *some kind of fruit*" or that "Ann likes *fruits similar to apples*". This former would require to provide a comprehensive list of every fruit while the latter would require the user to specify all fruits he deems similar to apples.

(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

Utilizing some form of SR/SS regarding to edge content for the matching step would allow users to define more generalising patterns. There exists a great variety of approaches for determining the SR/SS of text, which can generally be divided into *Corpus-based Measures* and *Knowledge-based measures* (Harispe et al. 2015, Section 1.3.2). The latter approaches may generally provide the explicitness in the measurement determination that is desired by CSS researchers. However among the former recent 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 Semantic Similarity* (NESS) measures in the following.

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 allow to specify matching edge content beyond isolated word semantics, although this may not always be desirable or necessary as in the example above.

As illustrated earlier, NESS measures principally do not provide the explicitness that is inherent to the pattern matching process of the SH framework. In the sense of the adaptive-open classification described above an integration of NESS would mean a shift from openness to adaptivity in this regard. While the SH framework generally can be situated in the realm of symbolic approaches, this integration would build a bridge between it and the realm of probabilistic approaches.

3.1 Research Questions

Based on the problem statement outlined above, we pose the following research questions:

3.1.1 Primary Question

R Can neural embedding-based semantic similarity regarding edge content be 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?

3.1.2 Secondary Questions

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 performance (recall) and how does it impact precision 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 Solution Approach

In this chapter we present the approach that was developed to answer the research questions (see section 3.1). Therefore trying to provide a solution to the problem of extending the SH framework by Neural Embedding-based Semantic Similarity Matching, which is described in chapter 3 where the relevancy of this problem for has also been derived.

The system is described here will in the following be referred to as *Neural Embedding-based Semantic Similarity extended Semantic Hypergraph Pattern Matching* or:

NESS-SHPM aka ***NESSeSHyPaM***

4.1 Integration into the Pattern Matching Process

4.1.1 `semsim` Functional Pattern

pattern works only for atoms

4.1.2 Sub-pattern Similarity Thresholds

4.2 Fixed Word Embedding-based Matching

word2vec via gensim

discussion about using transformer models for single word embeddings?

single-word and multi-word reference

Square bracket notation

4.3 Contextual Embedding-based Matching

Contextual Neural Embedding-based Semantic Similarity (CNESS)

i generally like your idea of contrasting the discrete and continous space as it allows to point out that there cant be one single point, also for a set of words which represents the meaning, but rather some subspace depending on the specific context Regarding the point of the semantic entities in continuous space being either word- or phrase based, the important difference is, that in case of semsim with context we do not compare the embedding representation of the phrases themselves. rather the sententences/phrases influence the embedding representations of the word (or maybe phrases) I tend to see this a bit like a blurring algo. The meaning of each token starts bleeding into its neighbours.

4.3.1 Context References

4.3.2 Token Mapping

4.4 Similarity Threshold Control

4.4.1 Breakpoint Discovery

detect change points in number of matches
see https://github.com/deepcharles/ruptures

half-max point and quarter/three-quarter points (percentiles, not quantiles) fit function and search for infliction as well as maximum derivative points, problematic in cases with less continuous change in number of matches.

how to approach this for practical applications?

5 Implementation

5.1 Relevant external Software Libraries used

Here list libs and models to be referenced later.

Word2Vec Gensim SentenceTransformers Transformers SpaCy

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 Evaluation and Results

In this chapter the conceived concept (see chapter 4) and specific implementation (see chapter 5) of the NESS-SHPM system is being evaluated to answer the research question(s) posed in section 3.1. Therefore a a case study is conducted to evaluate the system for a specific use case. In this case study quantitative results as well as qualitative examinations of the behaviour of NESS-SHPM are conducted. The quantitative results display the systems performance using metrics which are established for retrieval and classification tasks. The qualitative results exemplary showcase detailed aspects of the systems behaviour in the given use case.

refer to the RQs more specifically?

6.1 Case Study: Conflicts

The conflicts case study follows the approach presented in (Menezes and Roth 2021, p. 22) where expressions of conflict are extracted from a given SH using a single pattern. In their work they build upon the information extracted by the pattern (e.g. the two conflict parties) to conduct further analysis, which are not in the scope of this work. Here the evaluation is limited to the task of classifying whether the content of a given edge in the SH is an expression of conflict or not. Or framed differently the task is to to retrieve exactly all those edges whose content is an expression of conflict.

should I explain why specifically the conflicts and not some other case study (i.e. dataset) –> because there was none... but then I need to show why there was none and what are the criteria for a case study to be suitable to evaluate the system

6.1.1 Reddit Worldnews Corpus

The corpus from which those expressions of conflict are retrieved consists of news titles that were shared on the social media platform *Reddit*. Specifically all titles shared between January 1st, 2013 and August 1st, 2017 on *r/worldnews*, which is described as: "A place for major news from around the world, excluding US-internal news."[1] This corpus contains 479,384 news headers and is in the following referred to as the *Worldnews-Corpus*.

Each of these headers is comprised of a single sentence and is represented as a sequence root edge in the SH constructed from it. In the following this SH is referred to as the *Worldnews-SH*. Parsing errors that may potentially occur during this constructed and can obstruct a correct retrieval of a wrongly parsed edge i.e. wrongly represented sentence. These errors are out of scope of this work. All edges in the Worldnews-SH are assumed to be correctly parsed.

refer to examples

[1] `http://reddit.com/r/worldnews`

6.1.2 Original Conflict Pattern

Pattern 3 is used in (Menezes and Roth 2021, p. 22) to extract conflicts between two parties SOURCE and TARGET, potentially regarding some TOPIC.

(PRED/P.so,x SOURCE/C TARGET/C [against,for,of,over]/T TOPIC/[RS]) ∧
(lemma/J >PRED/P [accuse,arrest,clash,condemn,kill,slam,warn]/P)

Pattern 3: Original conflict pattern

6.1.3 SemSim Conflict Patterns

The part of Pattern 3 that mostly defines its retrieval performance is the list of predicates. This list has been manually constructed by (Menezes and Roth 2021) by investigating the Worldnews-Corpus. To evaluate the NESS-SHPH, this pattern will be modified to include the semsim functional pattern instead of the lemma functional pattern and the fixed word list. The general form of of such a pattern is shown by Pattern 4. This means only the matching of the predicate will be subject NESS, which allows to isolate the effects of the integration of NESS in the pattern matching process.

add proof / restructure this

(PRED/P.so,x SOURCE/C TARGET/C [against,for,of,over]/T TOPIC/[RS]) ∧
(semsim/J PRED/P .../P)

Pattern 4: General SemSim pattern

6.1.4 Edge predicate lemmas

Every matching edge in the full dataset contains a predicate sub-edge, which will be assigned to the PRED variable. Everyone of those sub-edges has an innermost atom which is a verb that has a lemma. In the following this will be called the *predicate lemma* of an edge. Each of the edges of Pattern 3 or a pattern in the form of Pattern 4 therefore corresponds to a predicate lemma.

6.1.5 Conflict Dataset

To conduct an evaluation which assesses the retrieval performance of the NESS-SHPM system it is necessary to have a dataset of edges with labels that state whether an edge is an statement of conflict or not. Since such a dataset did not exists it needs to be constructed. In the following the characteristics and the construction process of the dataset which was used for the evaluation in this case study will be shown.

Definition of Conflict

An expression of conflict in the context of this case study is defined as a sentence which fulfils the following properties:

There is a conflict between two explicitly named actors, wherever these actors are mentioned in the sentence; whereby a conflict is defined as antagonizing desired outcomes.

Desired Characteristics

To effectively evaluate the effectiveness of the application of NESS by matching a pattern in the form of Pattern 4, the dataset used for this should have the following characteristics:

- Contain the largest possible number of unique predicate lemmas
- Contain the largest possible number of edges per unique predicate lemma

On the one hand it is desired to have as many different unique predicate lemmas as possible in the dataset to be able to evaluate whether NESS can differentiate if a predicate lemma indicates an expression of conflict or not. On the other hand it is desired to have as many different edges per unique lemma as possible in the dataset to be able to evaluate whether CNESS is able to differentiate if edges represent an expression of conflict or not, given that they correspond to the same predicate lemma.

Construction Process

The set of edges that can be retrieved by a pattern following the form of Pattern 4 is restricted the general structure of this pattern. That means every set of matching edges for a pattern of this form will be a subset of the matching edges of the pattern which uses a wildcard operator to match the predicate (see Pattern 5). The results of matching this pattern against the Worldnews-SH are therefore used as the basis set of edges from which the conflict dataset is constructed, instead of the entirety of all its sequence root edges.

(PRED/P.so,x SOURCE/C TARGET/C [against,for,of,over]/T TOPIC/[RS]) ∧
(PRED/P */P)

Pattern 5: Wildcard predicate pattern

Matching Pattern 5 against the Worldnews-SH results in $n_f = 69380$ matching edges. In the following the set of those edges will be referred to as the *full* dataset.

Sampling Due to the excessive time required for labelling n_f edges and the limited availability of just three annotators, the full dataset needs to be subsampled to create the labelled dataset. The edges in the full dataset correspond to $n_l = 1800$ unique predicate lemmas. Attaining to the desired dataset characteristics, the number of samples n_s in the subsampled dataset should ideally be a multiple $m_l >= 2$ of n_l, so that $n_s = m_l \cdot n_l$. This would mean that every predicate lemma contained in the full dataset is statistically represented multiple times in the subsampled dataset.

exact number

A dataset size of $n_s = 2000$ was chosen, wich means $m_l < 2$ and $n_s << n_f$. This entails that a trade-off between the desired dataset characteristics has to be made. To account for this, a sampling method is applied that offers more control over the distribution of predicate lemmas in the subsampled dataset than uniform random sampling does. This sampling method is based on the idea of *Stratified Sampling* (Parsons 2017) and is described

is this correct?

in detail in algorithm 1.

The procedure splits the full dataset into multiple bins after the edges are sorted by number of occurrence of their predicate lemma and then uniformly randomly samples from each bin. This method guarantees that predicate lemmas which correspond to a relatively small number of edges in the full dataset will be represented in the subsampled dataset, while still representing the distribution of the full dataset.

do I have to show this in some way? should I then mention it at all?

Algorithm 1 Dataset sampling algorithm

1. Create a list of tuples t of edges and their corresponding predicate lemma: $L = [(l_k, e_i), ...]$ with $k \in \{0, ..., m\}$ and $i \in \{0, ..., n\}$
2. Sort this list by the number of tuples containing a predicate lemma to create the list: $L_{sort} = [(l_0, e_0), ...(l_m, e_n)]$, so that:
   - n_k is the number of tuples containing a lemma l_k
   - t_j with $j > i$ is a tuple wich sorted after tuple t_i
   - $n_o >= n_p$ if $t_i = (l_o, e_i)$ and $t_j = (l_p, e_j)$
3. Split the list L_{sort} into n_b bins.
4. Uniformly sample n_{sb} items from each bin.
5. Build a set of all edges e contained in the sampled tuples.

The resulting dataset size is $n_s = n_b * n_{sb}$. Given $n_s = 2000$, the values $n_b = 10$ and $n_{sb} = 200$ were chosen for sampling the dataset that is labelled.

Labelling The labelling task is shared between the three annotators. A given edge will be either labeled as *conflict* or *no conflict* by an annotator following the definition given above. Because of the aforementioned time constraints, every edge is only labeled by one annotator. To nonetheless ensure a consistent labelling among all annotators, a set of 50 edge is labelled by all three annotators. Every edge for which a disagreement in labelling occurs between at least two of the annotators, is inspected to reach an agreement on the label. Utilizing this process, the annotators understanding of what constitutes an expression of conflict is refined. Following this preliminary step, the n_s edge of the dataset are equally distributed among the three annotators and individually labelled by them.

Dataset Description

6.1.6 Evaluation Process

Evaluation Patterns

Dataset name	Number of all edges	Number of conflict edges (% of all edges)	Number of no conflict edges (% of all edges)
Worldnews-SH	479384	-/-	-/-
pred_wildcard_full	69380	-/-	-/-
pred_wildcard_subsample-2000	2000	599 (29.95 %)	1401 (70.05 %)

Table 6.1: Dataset descriptions

Pattern name	Lemma based	SemSim type	Requires ref. words	Requires ref. edges
Original conflict pattern	Yes	-/-	-/-	-/-
pred_semsim-fix_wildcard	No	FIXED	Yes	No
pred_semsim-fix-lemma_wildcard	Yes	FIXED	Yes	No
pred_semsim-ctx_wildcard	No	CONTEXT	No	Yes

Table 6.2: Evaluation patterns

7 Conclusion

8 Future Work

8.1 Implementation Improvements

implemnt multiprocessing, i.e. server process for both hypergraph and semsim matchers. other option would be to leverage python shared memory capabilities but is likely to be less stable and has less scaling potential

8.2 Further Evaluations

Bibliography

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).

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).

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.

Parsons, Van L. (2017). "Stratified Sampling". In: *Wiley StatsRef: Statistics Reference Online*. John Wiley & Sons, Ltd, pp. 1–11. ISBN: 978-1-118-44511-2. DOI: 10.1002/9781118445112.stat05999.pub2. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118445112.stat05999.pub2 (visited on 01/11/2024).