Edge Labelling in Narrative Knowledge Graphs

Vani Kanjirangat*, Alessandro Antonucci

Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA), USI-SUPSI, Lugano, Switzerland

Abstract

Edge labelling represents one of the most challenging processes for knowledge graph creation in unsupervised domains. Abstracting the relations between the entities, extracted in the form of triplets, and assigning a single label to a cluster of relations might be quite difficult without supervision and tedious if based on manual annotations. This seems to be particularly the case for applications in literary text understanding, which is the focus of this paper. We present a simple but efficient way to label the edges between the character entities in the knowledge graph extracted from a novel or a short story using a two-level clustering based on BERT-embedding with supersenses and hypernyms. The lack of benchmark datasets in the literary domain poses significant challenges for evaluations. In this work-in-progress paper, we discuss preliminary results to understand the potential for further research.

Keywords

Edge Labels, Verb Clusters, Supersenses, Lowest Common Hypernyms, Knowledge Graphs.

1. Introduction

Extracting structured information from narrative texts is a significant challenge for contemporary AI. The complexity further increases in the case of literary text because of possible ambiguous usage of words, neologisms, unique author writing styles, and many other subtle linguistic aspects. In fact the analysis of literary texts involves various complex steps such as the identification of the main characters and relations and their *typification* (e.g., gender, partnerships, goodness). Moreover, the high variance in style and the lexicon with frequent use of neologisms [1] and figures of speech [2] further complicates the scenario. Most of the past explorations are limited to particular application areas, such as biomedical literature [3, 4], or news and social media analysis [5, 6]. Different embedding techniques and the more recent attention based models, including transformers, evolved as the state-of-the-art for both unsupervised and supervised NLP tasks [7, 8, 9, 10, 11, 12, 13].

Identifying a more abstract and meaningful edge label for unsupervised knowledge graph extractions and its evaluation is a challenging process. We report here the current state of our work in the field with preliminary experiments on unsupervised edge labelling of knowledge graphs extracted from literary texts. A simple technique to label the edges in a reasonable way is evaluated. The code is already available in a public repository (github.com/IDSIA/novel2graph).

In: R. Campos, A. Jorge, A. Jatowt, S. Bhatia, M. Litvak (eds.): Proceedings of the Text2Story'23 Workshop, Dublin (Republic of Ireland), 2-April-2023

*Corresponding author.

vanik@idsia.ch (V. Kanjirangat); alessandro@idsia.ch (A. Antonucci)

2. Related Work

The onset of deep learning has given the drive to powerful data processing models to ease NLP applications. For *knowledge graphs* (KGs), deep models are used to embed the triplet information and address tasks such as link predictions and graph completion [14, 15] and training embeddings [16, 17, 18, 19]. Another major shift was the introduction of attention, and transformer models [20], with many works that adopting attention mechanisms for KG completion and learning tasks [21, 22, 23]. There has been also focus towards unsupervised learning of KG embeddings [24, 25].

The automatic interpretation and visual analysis of literary texts have been explored from various perspectives in the past few years. In [26], the literary characters and the network associations have been studied, while in [27] sentiment relations between (Shakespeare's) characters have been processed. When it comes to unsupervised KG constructions, a combination of classical and deep learning NLP techniques is usually required.

3. Methods

Let us first briefly discuss the entity extraction process, which is a necessary preprocessing already studied in our previous works before approaching the edge labelling approach. Since, we are dealing with literary text, our entities are the characters in the given input novel or short story, which exhibit various characteristics and relations. As in [28], we used the Stanford *Named Entity Recognition* Tagger[1] together with a character *de-aliasing*, i.e., unifying the character names that can be possibly referred in different ways (e.g., *Ron* and *Ronald*). This is achieved by the DBSCAN clustering algorithm [29] paired with the Levenshtein string distances. We use the *partial_ratio* method provided by the *fuzzywuzzy* module[2] to compute the distance matrix. This is followed by the coreference resolution[3] using the Stanford package and some heuristic adjustments. Each character entity is eventually represented by a unique identifier. These entities define the nodes of the KG. The next step is to label the edges connecting these nodes, which is the major focus of the present work.

3.1. Verb Extraction and Embedding

Following [28], we extract all the sentences containing two characters/entities and exclude self-relations (e.g., *Harry, I am Harry Potter*). For simplicity, we also prune the sentences, where the second character appears at the end of the sentence (e.g., *said Harry*). To split the larger sentences, we use constituency parsing tree[4] to extract the subtrees. Our approach traverses the tree using a depth-first search and extracts each phrase (S) containing at least one noun phrase (NP) and one verb phrase (VP) starting from the bottom of the tree. For instance, consider the sentence:

CHAR0 is talking to CHAR1, while CHAR1 is cooking for CHAR2.

[1] https://nlp.stanford.edu/software/CRF-NER.html
[2] https://pypi.org/project/fuzzywuzzy
[3] https://nlp.stanford.edu/projects/coref.shtml
[4] https://stanfordnlp.github.io/CoreNLP/parse.html

The constituency parsing tree returns two extracted phrases (*CHAR0 is talking to CHAR1* and *CHAR1 is cooking for CHAR2*) as depicted in Fig. 1.

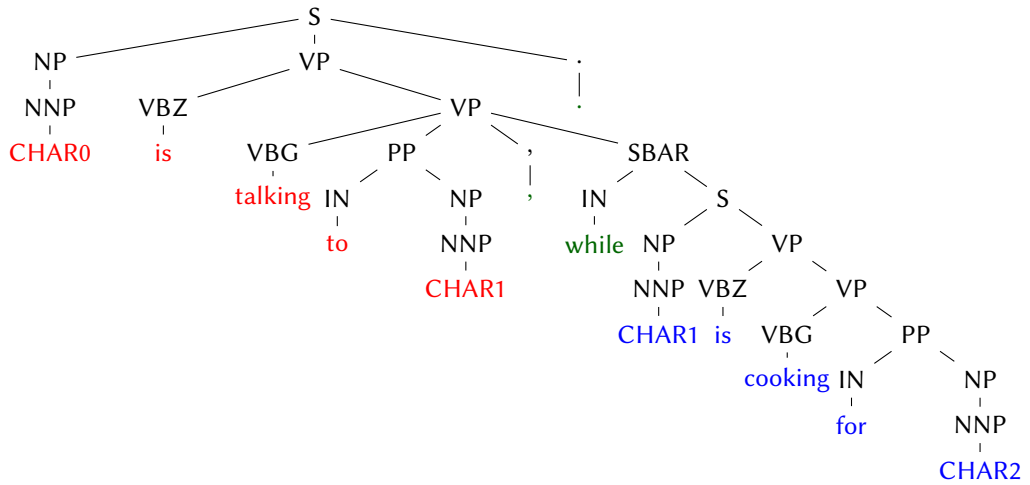

Figure 1: Phrase segmentation based on constituency parsing tree.

We refer to the set of output sentences as *relational sentences*. Once we have all the relational sentences, the next step is to extract a representative verb for each relational sentence. Using Part-of-Speech (POS) tagging, we extract the verbs in these relational sentences. Further, we embed the sentences using Sentence BERT (SBERT) [30] and extract the embeddings of the corresponding extracted verbs. SBERT uses a Siamese network structure [31] to produce meaningful sentence encodings. Once we have the embedded verbs, we group similar verbs together. Since the embeddings are supposed to encode semantic or contextual information, sentences with similar vector representations are supposed to share similar relations.

3.2. Verb Clustering and Edge Labelling

Algorithm 1: Verb Clustering (Level 1)

Input: Extracted Verbs [V_1,V_2,..V_n], Supersense Categories (SC)
Output: Supersense-based Verb Clusters

```
Find embeddding of [V_1,V_2,..V_n];
if Verb in single SC then Assign it to that SC;
else if Verb in multiple SCs then
    for SC do
        Remove the verb from SC;
        Compute average embedding of SC with the remaining verbs;
if Verb not in any SC then Compute the average embedding of SCs;
Compute distance between the verb embedding and the average embeddings of SCs;
Assign the verb to the SC at minimum distance;
```

Algorithm 2: Verb Clustering (Level 2)

Input: Supersense-based Verb Clusters
Output: Triplet ($C1$, r,$C2$)

```
for each supersense-based verb cluster do
    Take all the verb pairs
for each verb pair do
    Compute the lowest common hypernym (LCH) and store them all;
    Sort the LCHs based on their frequency;
for each verb do
    Associate it to the most common LCH;
    if no LCH associated to a verb then Consider as outlier;
    Associate the relation label with the corresponding LCH;
    Generate the triplets;
```

To achieve this, we adopt the two-level verb clustering summarised by Algs. 1 and 2. The first step involves grouping the extracted verbs into *supersense* clusters as given in Alg. 1. Supersense (SS) [32] is a terminology from WordNet [33], where the words are grouped into sets of synonyms called *synsets*. Each synset is associated with one of the 45 broader semantic categories/SSs, out of which we have 26 nouns, 16 verbs, 3 adjectives, and 1 adverb. This can be regarded as a coarse-grained word sense grouping, but it can be quite helpful for many NLP tasks. We focus on verb SS category only, as we consider the verbs in a sentence as the input. A word can belong to multiple SS categories (as a word can have different senses), and hence SS tagging or disambiguation is another challenging research problem. In the proposed approach, we consider the 16 verb SSs as the category or clusters to which an input verb has to be assigned, which are *{body, change, cognition, communication, competition, consumption, contact, creation, emotion, motion, perception, possession, social, stative, weather}*. We then compute the embeddings of the extracted verbs with SBERT. Further, we follow the steps from 2 to 8 in Alg. 1 to assign the verb to a specific SS category. If the verb belongs to multiple SS category or to none of these categories, we compute the average of all verb embeddings that belong to each SS category and assign the verb to one with which it has minimum cosine distance.

The input to second-level, described in Alg. 2 is the SS-based verb clusters. We take all the verb pairs in a cluster and compute the *lowest-common-hypernyms* (LCHs), which is the lowest common ancestor node between the given synsets in the hierarchy. Since each verb can have multiple synsets, we can have multiple LCHs for a verb pair. These are then sorted based on the frequency of their occurrences, which are related to the strength of association with the verb pair and associate it to the most common LCH. This LCH is considered as the edge label and we generate the triplets ($C1$, r,$C2$), where r is the predicate/relation and $C1$ and $C2$ are the entities/characters. E.g., for the verb cluster *{call, pass, share , give, take, spend, buy}*, the output is *{Synset('move.v.02'): ['save', 'call', 'pass', 'give', 'take'],Synset('act.v.01'): ['share'],Synset('give.v.03'): ['spend', 'buy']}*.

4. Experiments

We use the first six books of the Harry Potter series by J.K. Rowling (885‘943 words). Tab. 1 shows the statistics of the number of sentences extracted before and after co-referencing for the first book. K-means with cosine distance is used for sentence clustering. Algs. 1 and 2 are applied. A snapshot of the supersense-based clusters obtained using the proposed approach defined is in Tab. 2 (left), while the final triplets obtained from verb clusters at level 2 is in Tab. 2 (right). Semantically similar verbs are properly clustered together under the corresponding supersense category. E.g., for category *communication*, we have verbs such as *speak*, *raise*, *warn*, and *mutter*. They are closely related to each other in the sense that all these verbs a different ways of communications to express the emotions and further character relations. The preliminary experiments show that our approach yield meaningful clusters and triplets.

Table 1
Statistics after different steps of relational sentence detection.

Type of Sentences	# Before/After Co-Referencing
Identified sentences	6394/6394
With two chars	566/618
Asymmetric sentences	511/564
Two different chars	470/516
Not included sentences	470/516
Not “... said charX... ”	387/433
Verb between chars	331/380

Table 2
Supersense category and verb clusters (left), representative verbs and triplets (right).

Verb Category	Verbs
stative	{shake,lose,study,relax,favor}
communication	{speak,raise,bully,cheer,warn,mutter}
consume	{growl,scramble,eat}
motion	{move,walk,slip,look}
emotion	{fuss,cast,recognize,scare}
possession	{hand,find,clap,save,borrow,award,swap}
body	{smile,laugh,grin,blink,spit}
perception	{fill,whip,fight,insist,glance,throw,break}
cognition	{snore,hear,feel,help,share,gasp,linger,dance}
social	{celebrate,dare,punish}

Verbs	Triplets
play, act	(Harry,play,Slytherin)
	(Harry,act,Snape)
complain, mutter	(Harry,mutter,Snape)
	(Ron,mutter,Harry)
block,fight	(Marcus,block,Harry)
	(Granger,fight,Snape)
say,repeat	(Quirrell,say,Snape)
	(Ron,repeat,Hagrid)

5. Conclusion

We described our preliminary experiments with an unsupervised edge labelling approach for knowledge graphs. A two-level clustering approach, based on verb supersenses and lowest common hypernyms has been used. To capture semantic similarity, we used the BERT-based embeddings. The approach was empirically evaluated on a literary text. As future work, we aim to enhance sense clustering by approaches such as sense-BERT [34].

References

[1] M. Martínez Carbajal, et al., Neologisms in Harry Potter books, Universidad de Valladolid. Facultad de Filosofía y Letras (2014).

[2] Å. Nygren, Essay on the linguistic features in J.K. Rowling's Harry Potter and the Philosopher's Stone, 2006.

[3] Y. Zhang, H. Lin, Z. Yang, J. Wang, S. Zhang, Y. Sun, L. Yang, A hybrid model based on neural networks for biomedical relation extraction, Journal of Biomedical Informatics 81 (2018) 83–92.

[4] X. Lv, Y. Guan, J. Yang, J. Wu, Clinical relation extraction with deep learning, International Journal of Hybrid Information Technology 9 (2016) 237–248.

[5] L. Q. Trieu, H. Q. Tran, M.-T. Tran, News classification from social media using Twitter-based doc2vec model and automatic query expansion, in: Proceedings of the Eighth International Symposium on Information and Communication Technology, 2017, pp. 460–467.

[6] S. Ghosh, C. Shah, Towards automatic fake news classification, Proceedings of the Association for Information Science and Technology 55 (2018) 805–807.

[7] O. Levy, Y. Goldberg, I. Dagan, Improving distributional similarity with lessons learned from word embeddings, Transactions of the Association for Computational Linguistics 3 (2015) 211–225.

[8] R. A. Stein, P. A. Jaques, J. F. Valiati, An analysis of hierarchical text classification using word embeddings, Information Sciences 471 (2019) 216–232.

[9] J. Wieting, M. Bansal, K. Gimpel, K. Livescu, Towards universal paraphrastic sentence embeddings, arXiv preprint arXiv:1511.08198 (2015).

[10] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), 2018, pp. 2227–2237.

[11] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[12] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, Q. V. Le, Xlnet: Generalized autoregressive pretraining for language understanding, in: Advances in Neural Information Processing Systems, 2019, pp. 5754–5764.

[13] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, arXiv preprint arXiv:1910.13461 (2019).

[14] B. Y. Lin, X. Chen, J. Chen, X. Ren, Kagnet: Knowledge-aware graph networks for commonsense reasoning, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 2822–2832.

[15] L. Yao, C. Mao, Y. Luo, KG-BERT: BERT for knowledge graph completion, arXiv preprint arXiv:1909.03193 (2019).

[16] G. Ji, S. He, L. Xu, K. Liu, J. Zhao, Knowledge graph embedding via dynamic mapping matrix, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing

(Volume 1: Long Papers), 2015, pp. 687–696.
[17] Z. Wang, J. Zhang, J. Feng, Z. Chen, Knowledge graph embedding by translating on hyperplanes, in: C. E. Brodley, P. Stone (Eds.), Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada, AAAI Press, 2014, pp. 1112–1119.
[18] Y. Lin, Z. Liu, M. Sun, Y. Liu, X. Zhu, Learning entity and relation embeddings for knowledge graph completion, in: B. Bonet, S. Koenig (Eds.), Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA, AAAI Press, 2015, pp. 2181–2187.
[19] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, in: Advances in Neural Information Processing Systems, 2013, pp. 2787–2795.
[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.
[21] X. Liu, H. Tan, Q. Chen, G. Lin, Ragat: Relation aware graph attention network for knowledge graph completion, IEEE Access 9 (2021) 20840–20849.
[22] C. Li, X. Peng, Y. Niu, S. Zhang, H. Peng, C. Zhou, J. Li, Learning graph attention-aware knowledge graph embedding, Neurocomputing 461 (2021) 516–529.
[23] H. Wang, S. Li, R. Pan, M. Mao, Incorporating graph attention mechanism into knowledge graph reasoning based on deep reinforcement learning, in: Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP), 2019, pp. 2623–2631.
[24] N. Sheikh, X. Qin, B. Reinwald, C. Miksovic, T. Gschwind, P. Scotton, Knowledge graph embedding using graph convolutional networks with relation-aware attention, arXiv preprint arXiv:2102.07200 (2021).
[25] N. Veira, B. Keng, K. Padmanabhan, A. G. Veneris, Unsupervised embedding enhancements of knowledge graphs using textual associations., in: IJCAI, 2019, pp. 5218–5225.
[26] A. Piper, M. Algee-Hewitt, K. Sinha, D. Ruths, H. Vala, Studying literary characters and character networks, 2017.
[27] E. T. Nalisnick, H. S. Baird, Character-to-character sentiment analysis in shakespeare's plays, in: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, volume 2, 2013, pp. 479–483.
[28] S. Mellace, V. Kanjirangat, A. Antonucci, Relation clustering in narrative knowledge graphs, in: Proceedings of AI4Narratives - Workshop on Artificial Intelligence for Narratives in conjunction with the 29th International Joint Conference on Artificial Intelligence and the 17th Pacific Rim International Conference on Artificial Intelligence (IJCAI 2020), Yokohama, Japan, volume 2794 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020, pp. 23–27.
[29] D. Birant, A. Kut, ST-DBSCAN: An algorithm for clustering spatial–temporal data, Data & Knowledge Engineering 60 (2007) 208–221.
[30] N. Reimers, I. Gurevych, Sentence-BERT: Sentence embeddings using Siamese BERT-networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 3973–3983.

[31] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: A unified embedding for face recognition and clustering, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 815–823.
[32] M. Ciaramita, M. Johnson, Supersense tagging of unknown nouns in wordnet, in: Proceedings of the 2003 conference on Empirical methods in natural language processing, 2003, pp. 168–175.
[33] G. A. Miller, WordNet: An electronic lexical database, MIT press, 1998.
[34] Y. Levine, B. Lenz, O. Dagan, O. Ram, D. Padnos, O. Sharir, S. Shalev-Shwartz, A. Shashua, Y. Shoham, Sensebert: Driving some sense into bert, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 4656–4667.