Temporal Embeddings and Transformer Models for Narrative Text Understanding

Vani K Simone Mellace Alessandro Antonucci

Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA)
Lugano (Switzerland)
{vanik,simone,alessandro}@idsia.ch

Abstract

We present two deep learning approaches to narrative text understanding for character relationship modelling. The temporal evolution of these relations is described by *dynamic word embeddings*, that are designed to learn semantic changes over time. An empirical analysis of the corresponding character trajectories shows that such approaches are effective in depicting dynamic evolution. A supervised learning approach based on the state-of-the-art transformer model BERT is used instead to detect static relations between characters. The empirical validation shows that such events (e.g., two characters belonging to the same family) might be spotted with good accuracy, even when using automatically annotated data. This provides a deeper understanding of narrative plots based on the identification of key facts. Standard clustering techniques are finally used for character *de-aliasing*, a necessary pre-processing step for both approaches. Overall, deep learning models appear to be suitable for narrative text understanding, while also providing a challenging and unexploited benchmark for general natural language understanding.

1 Introduction

Due to the inherent complexity involved in textual data, narrative text understanding remains a challenging and relatively unexplored research area for AI. Here we consider narrative text, such as novels and short stories (broadly termed here as *literary text*) and try to address its lexical diversity and richness in terms of relations between entities [PAHS+17]. In recent years, *Deep Learning* (DL) approaches were found to positively impact *Natural Language Processing* (NLP) with impressive boosts in text extraction and understanding capabilities. This marginally concerns the area of literary text [LB19], where the application of DL models remains relatively unexplored. Some researchers modelled character networks using machine learning, mostly from a social network perspective based on generative models for conversational dialogues [ACJR12, CHTH+10] not involving DL state-of-the-art approaches. Just a few works have been reported for character evolution and relational analysis [CSDID16, KA19, VKA20].

Here we evaluate the application of DL to literary text understanding. The goal is to describe character relationships within a novel and their evolution. Moreover, we also want to emphasize the potential of literary

In: R. Campos, A. Jorge, A. Jatowt, S. Bhatia (eds.): Proceedings of the Text2Story'20 Workshop, Lisbon, Portugal, 14-April-2020, published at http://ceur-ws.org

text as a challenging benchmark for state-of-the-art language models, whose major applications are typically in other domains such as biomedical literature [LZFJ17] or fake news detection [RSL17], where both the lexical richness and the intricacy of the inter-entities relations might be less intricate compared to literary domain.

To analyse the character relationships, both supervised and unsupervised DL techniques are considered here. A classification model to identify the relations between characters using BERT (*Bidirectional Encoder Representations from Transformers*, [DCLT19]) is trained from supervised data. BERT is successfully used in various classification tasks, but, to the best of our knowledge, not yet in the literary text domain. Moreover, manual annotation of training data in this field can be very expensive, this representing a strong limitation for this direction. To partially bypass this issue, here we also present a simple approach to automatically generate training data for character relation classification (focusing on family relations, such as *parent of*, *sibling of*).

At the unsupervised level, we consider the dynamic evolution of the characters over time (i.e., across the text). To do that, we learn vectors associated to different characters based on so-called *dynamic or temporal embeddings* [BM17], allowing to learn vectors over different slices inside the text (e.g., chapters or fixed amounts of text), while maintaining the vectors comparable over time because of a common initialization. We analyse the relations between characters by visualizing the character trajectories over time by low-dimensionality projections or the relative distances in the original, high-dimensionality, spaces or by considering the relative distances between the vectors.

Both techniques require a pre-processing step consisting in character detection, based on standard entity recognition techniques, and character de-aliasing, for which density-based clustering methods are adopted.

The paper is organized as follows. A review of existing work is in Section 2. Sections 3 and 4 report a discussion of, respectively, the supervised and unsupervised approaches. An empirical validation is in Section 5. Conclusions and outlooks are finally reported in Section 6.

2 Literature Review

The onset of DL has given drive to powerful data processing models, which facilitate NLP applications. In this context, systems that understand the semantic and syntactic aspects of a text are extremely important. Word embedding models such as *Word2Vec* [MSC$^+$13], or *Glove* [PSM14], as well as sentence embedding models such as USE (*Universal Sentence Encoder*, [CYK$^+$18]) help in representing text as a mathematical object in a reliable way. The text representations using such embeddings along with NLP and deep neural networks [HS97, MMY$^+$16] played a vital role in text extraction, classification and clustering.

Another major shift was the introduction of attention models and transformers [VSP$^+$17], these are language models able to better understand text semantics by contextual analysis. BERT, ELMO [PNI$^+$18] and various versions of these models gave a big boost to recent NLP applications [HSG$^+$19]. Moreover, word embeddings, originally intended as static model of a given corpus, later led to the exploration of their dynamic evolution over time, this being mainly used to compare the semantic shifts of words over time and detection of word analogies [KARPS15, KØSV18]. Notably, some of these works used BERT for story ending predictions and temporal event extractions [HLAP19, LDL19]. In the next sections, we show how these models can be applied to literary text understanding.

3 Character Trajectories by Temporal Word Embeddings

Both the unsupervised technique presented in this section and the supervised approach discussed in Section 4 require a reliable identification of the characters involved in the plot. This corresponds to a named entity recognition task, for which standard tools can be used.[1] As same characters can occur in the text with different aliases (e.g., *Ron* and *Ronald* Weasley), a *de-aliasing* might be needed as an additional pre-processing step. We achieve that by a clustering of the named entities based on the DBSCAN algorithm [BK07]. The entities are clustered using precomputed distances based on the *sequence matcher* algorithm, which finds the longest common subsequences.

After character identification and de-aliasing, learning the embeddings of the characters of a literary text is a straightforward task. As the learning of an embedding is based on contextual information, the only important condition is that a sufficient amount of co-occurrences of the characters in the text is available. If this is the case, the relative distances between the vectors can be used as proxy indicators of the relations between the corresponding characters. This can be also achieved for separate parts of a same text (e.g., chapters), provided

[1] E.g., see `https://nlp.stanford.edu/software/CRF-NER.html`.

that the amount of text remains sufficient for learning. In this way it is possible to capture the relations between characters for each part, but not to describe the dynamic evolution of the same character over the whole text. Vectors trained in different embeddings, even with the same dimensionality, are in fact not directly comparable.

The method employed in [DCBP19] elegantly addresses this issue by *aligning* different temporal representations using a shared coordinate system. The model uses a skip-gram Word2Vec architecture, where the context matrix (the output weight matrix) is fixed during the training, while allowing the word embedding input weight matrices to change on the basis of co-occurrence frequencies that are specific to a given temporal interval. After training, model returns the context embeddings, that we are going to consider as a *temporal* word embedding. To achieve that, first, a static word embedding is trained with random initialization using the whole text and ignoring temporal slices.

Let us denote as $\boldsymbol{W}$ the corresponding word embedding matrix and as $\boldsymbol{W}'$ the corresponding context matrix. For each slice, we instead initialize the word embedding matrix with $\boldsymbol{W}$ while keeping $\boldsymbol{W}'$ as a frozen context matrix equal for all the time slices [DCBP19]. This initialization has been proved to force alignment and make it possible to compare vectors from embeddings associated to different time slices. The architecture is depicted in Figure 1. In particular, we adopt the dynamic initialization scheme proposed in [VKA20], which appears to be more suitable for narrative text because of its intrinsic sequential nature.

Dynamic embeddings, generally used for word analogies, are considered here to describe and interpret relations by means of the trajectories spanned by the vectors associated to different characters. The character embeddings are represented in a visual space by dimensionality reduction [MH08] to understand the evolving relations between characters, in terms of time slices in a novel such as chapters or other parts of the text. This could be further related to character sentiments, clustering of emotions and other descriptions.

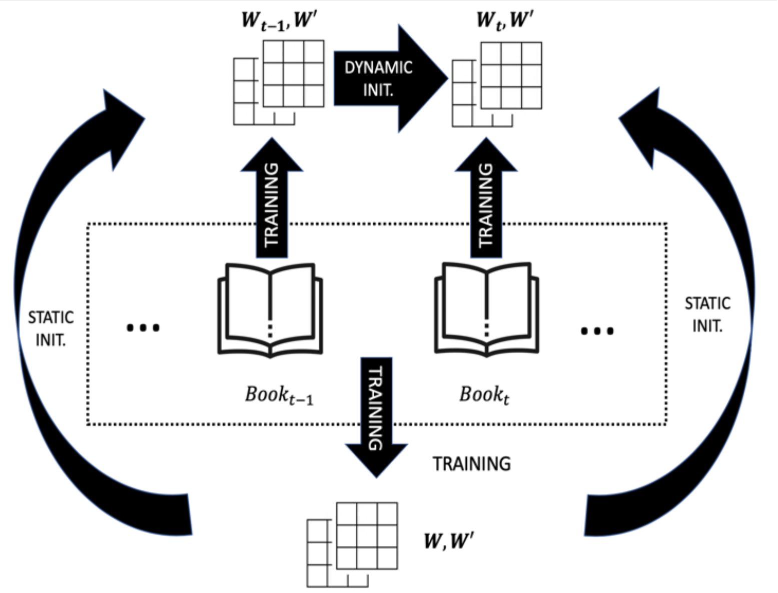

Figure 1: Training temporal embeddings

4 BERT-based Classification of Character Relations

The unsupervised approach considered in the previous section describes the relation between characters in terms of relative positions of the corresponding vectors and their evolution. Here we consider a character relation extraction based on binary classifiers. This is a supervised approach based on a ground truth of annotated sentences where the two characters are identified together with a Boolean value expressing whether or not the relation under consideration is met. The character names or aliases are eventually replaced by anonymous placeholders, as this helps the model to learn the relationships by abstracting from the specific names.

For the learning phase, we use the BERT classification model. Its pre-trained model can be fine tuned for classification with an additional output layer. BERT has a wordpiece tokenizer using two special tokens (SEP and CLS), which are used to encode valuable information of sentence structure and semantics after fine-tuning.

The BERT-base has twelve transformer layers and in the classification task, the pooled token embedding from the CLS tokens is fed into a linear classifier for predictions. With the powerful attention mechanisms, BERT embeddings encode deep semantic and syntactic contextual information. The relation extraction problem is modelled as a single sentence classification task using BERT model. More details about this general architecture are in [DCLT19].

As creating ground truth in this field might be very expensive, we also discuss techniques for automatic data annotation. As an example, let us focus on family relations, where the problem is to decide whether or not there is familial relation between two characters. By increasing the neighbourhood parameter, the output of the DBSCAN clustering algorithm used for de-aliasing produces clusters in which the characters belonging to same family are together (as their second names remain same). E.g., in the *Harry Potter* books, we have clusters with the *Potter* and the *Weasley* family. These clusters are used for automatic creation of the positive samples in training data, while the remaining characters are used for negative sample generation.

5 Experimental Analysis

The above discussed approaches to character relation modelling (Section 3) and understanding (Section 4) are validated here with two novels: *Little Women* (LW) by L.M. Alcott (text length 197‘524 words) and the first six books of the *Harry Potter* series (HP) by J.K. Rowling (885‘943 words).

Family Relationship Classification.

Due to the intricate nature of its plot and its length, HP is being often used as a benchmark for natural language understanding in literary domain [BDE+16, Spa13]. As an application of the ideas discussed in Section 4, we consider the task of predicting whether or not a given pair of characters has a family relation or not. A BERT based classifier is used for that.[2] Out of six books, we use the sentences generated from five books as training set and the remaining book as a test according to a cross-validation scheme. For the training set, the automatic class labelling is done by creating clusters for the same family groups (see Section 4). The number of samples for each book is 160, 250, 239, 396, 478, and 231, the ratio of positive samples for each book being 30.0%, 39.2%, 28.9%, 38.6%, 62.6%, and 47.6%. BERT is used together with the Adam optimizer [KB14]. This gives a learning rate equal to $2 \cdot 10^{-5}$, warm up equal to 0.1 and ten epochs. The results in Table 1 show reasonably good average performances and their standard deviations over the six books. Note that the aggregated values, corresponding to weighted averages, might be higher than those for negative or positive samples only.

Table 1: Character familial relation classification in *Harry Potter*

Samples	Precision	Recall	F-score
Negative	$79 \pm 13\%$	$85 \pm 7\%$	$81 \pm 8\%$
Positive	$77 \pm 9\%$	$71 \pm 9\%$	$73 \pm 4\%$
All	$80 \pm 4\%$	$78 \pm 7\%$	$78 \pm 7\%$

A test is also done on the LW benchmark with the HP training data. A lower F-score level (64%) is obtained, possibly related to an over-fitting effected. This might be relevant for literary texts, where the differences between different data (e.g., different texts of different authors) are typically stronger than in other domains.

It is important to note we have implemented a classification model whose predictions are at the sentence level. When coping with character pairs, it would be more appropriate to consider a higher level, i.e, prediction with respect to all the sentences that express the character pair relation. This is achieve by a *bag of sentence* approach, where a character pair is considered to have a relation, if at least one of the sentences is predicted as positive. For HP there are 85 entity pairs (12 positive and 73 negative) and the results are 9 positive (75%) and 66 (90%) negative pairs correctly predicted. Considering the intrinsic complexity of literary text, where sentences might have very complex structures, this might be regarded as a promising result and also advocate our strategy for the automatic generation of training set.

[2] See `https://github.com/huggingface/transformers`.

Temporal Word Embeddings.

Following the discussion in Section 3, we train a temporal word embedding for the first six books of HP. We focus on the four characters which appear more frequently. The static embedding is trained with the whole text of each book, while the dynamic embeddings are based on sub-slices containing text of length equal to 1000 characters. For each character, we extract the corresponding trajectory for each book. For a better interpretation of the relations, we consider the main character (i.e., *Harry*) and plot the evolution over time of the relative (cosine) distances from the other characters. Since these vectors embed semantic information, it is expected that in the trajectories corresponding to smaller distances correspond to closer relations with Henry. In fact, the results in Figure 2 show that the trajectories of positive characters or friends (i.e., *Ron* and *Hermione*) move in a similar way. The main antagonist (i.e., *Voldemort*) is found instead to move in a different direction and at a higher distance.

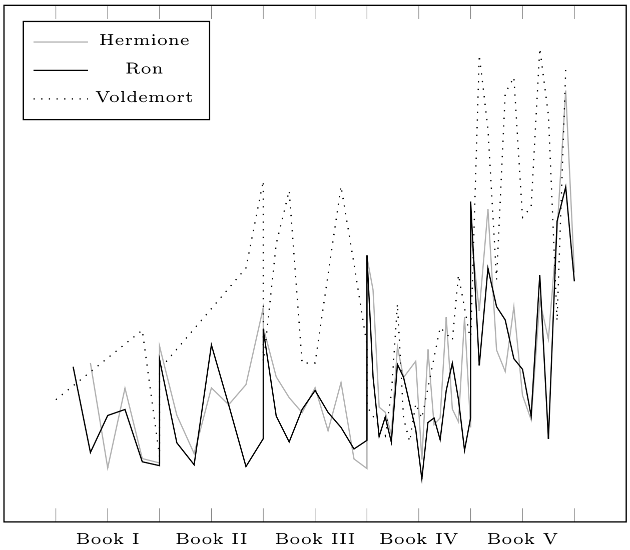

Figure 2: Characters trajectories for *Harry Potter*

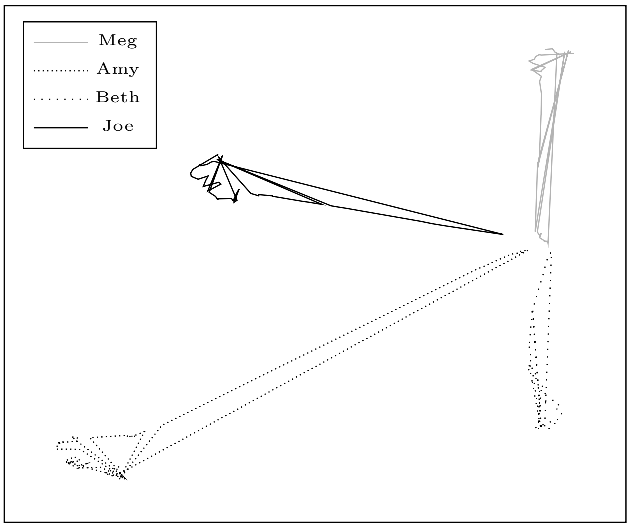

Figure 3: Character trajectories for *Little Women*

A similar analysis for LW is reported in Figure 3. In this case we display a t-SNE [MH08] two-dimensional projection of the vectors over different groups of chapters for the four major characters (i.e., the four *March* sisters). As a comment, the temporal word embedding seems to capture the separation, during the central part of the plot, between *Joe* and *Amy*, i.e., the two characters who left their home town, and the other two.

6 Conclusion

In this paper, we presented supervised and unsupervised DL models for analysing and interpreting character relations in a novel. We used BERT classifiers for predicting the character relations, while an unsupervised approach based on temporal word embeddings was used to interpret the character relation evolution. Both methods are found to be promising to explore the relations involved within characters in a novel. Thus, the approaches can be further applied to literary text understanding for deriving character networks and hence studying the relations and sentiments involved. In future, we want to integrate these approaches to build a more user-friendly tool to analyse the character networks and use it for an extensive validation.

References

[ACJR12] Apoorv Agarwal, Augusto Corvalan, Jacob Jensen, and Owen Rambow. Social network analysis of Alice in Wonderland. In *Proceedings of the NAACL-HLT 2012 Workshop on computational linguistics for literature*, pages 88–96, 2012.

[BDE+16] Anthony Bonato, David Ryan D'Angelo, Ethan R Elenberg, David F Gleich, and Yangyang Hou. Mining and modeling character networks. In *International workshop on algorithms and models for the web-graph*, pages 100–114. Springer, 2016.

[BK07] Derya Birant and Alp Kut. ST-DBSCAN: An algorithm for clustering spatial–temporal data. *Data & Knowledge Engineering*, 60(1):208–221, 2007.

[BM17] Robert Bamler and Stephan Mandt. Dynamic word embeddings. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 380–389, 2017.

[CHTH+10] Asli Celikyilmaz, Dilek Hakkani-Tur, Hua He, Greg Kondrak, and Denilson Barbosa. The actor-topic model for extracting social networks in literary narrative. In *NIPS Workshop: Machine Learning for Social Computing*, page 8, 2010.

[CSDID16] Snigdha Chaturvedi, Shashank Srivastava, Hal Daume III, and Chris Dyer. Modeling evolving relationships between characters in literary novels. In *Proceedings of AAAI*, 2016.

[CYK+18] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, 2018.

[DCBP19] Valerio Di Carlo, Federico Bianchi, and Matteo Palmonari. Training temporal word embeddings with a compass. In *Proceedings of AAAI*, volume 33, pages 6326–6334, 2019.

[DCLT19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[HLAP19] Rujun Han, Mengyue Liang, Bashar Alhafni, and Nanyun Peng. Contextualized word embeddings enhanced event temporal relation extraction for story understanding. *arXiv:1904.11942*, 2019.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[HSG+19] Hebatallah A. Mohamed Hassan, Giuseppe Sansonetti, Fabio Gasparetti, Alessandro Micarelli, and J. Beel. BERT, ELMo, USE and InferSent sentence encoders: The panacea for research-paper recommendation? *CEUR Workshop Proceedings*, 2431:6–10, 2019.

[KA19] Vani K and Alessandro Antonucci. Novel2graph: Visual summaries of narrative text enhanced by machine learning. In Mário Jorge Alípio, Campos Ricardo, Jatowt Adam, and Bhatia Sumit, editors, *Proceedings of Text2Story - Second Workshop on Narrative Extraction From Texts co-located with 41th European Conference on Information Retrieval (ECIR 2019)*, pages 29–37. CEUR Workshop Proceedings, 2019.

[KARPS15] Vivek Kulkarni, Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. Statistically significant detection of linguistic change. In *Proc. of the 24th Int. Conf. on World Wide Web*, pages 625–635, 2015.

[KB14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[KØSV18] Andrey Kutuzov, Lilja Øvrelid, Terrence Szymanski, and Erik Velldal. Diachronic word embeddings and semantic shifts: a survey. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1384–1397, 2018.

[LB19] Vincent Labatut and Xavier Bost. Extraction and analysis of fictional character networks: A survey. *ACM Computing Surveys (CSUR)*, 52(5):1–40, 2019.

[LDL19] Zhongyang Li, Xiao Ding, and Ting Liu. Story ending prediction by transferable BERT. *arXiv:1905.07504*, 2019.

[LZFJ17] Fei Li, Meishan Zhang, Guohong Fu, and Donghong Ji. A neural joint model for entity and relation extraction from biomedical text. *BMC bioinformatics*, 18(1):198, 2017.

[MH08] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[MMY$^+$16] Lili Mou, Zhao Meng, Rui Yan, Ge Li, Yan Xu, Lu Zhang, and Zhi Jin. How transferable are neural networks in nlp applications? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 479–489, 2016.

[MSC$^+$13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *NIPS*, pages 3111–3119, 2013.

[PAHS$^+$17] Andrew Piper, Mark Algee-Hewitt, Koustuv Sinha, Derek Ruths, and Hardik Vala. Studying literary characters and character networks. In *DH*, 2017.

[PNI$^+$18] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237, 2018.

[PSM14] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[RSL17] Natali Ruchansky, Sungyong Seo, and Yan Liu. Csi: A hybrid deep model for fake news detection. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 797–806, 2017.

[Spa13] Amelia Carolina Sparavigna. On social networks in plays and novels. *International Journal of Sciences*, 2(10), 2013.

[VKA20] Claudia Volpetti, Vani K, and Alessandro Antonucci. Temporal word embeddings for narrative understanding. In *ICMLC 2020: Proceedings of the Twelfth International Conference on Machine Learning and Computing*, ACM Press International Conference Proceedings Series. ACM, 2020. ISBN: 978-1-4503-7642-6.

[VSP$^+$17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017.