AN AUTOMATED DOMAIN UNDERSTANDING TECHNIQUE FOR KNOWLEDGE GRAPH GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

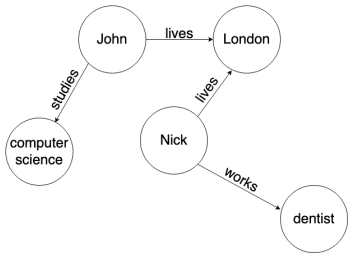
Domain-specific Knowledge Graph (KG) generation is a labor intensive task usually orchestrated and supervised by subject matter experts. Herein, we propose a strategy to automate the generation process following a two-step approach. Initially, the structure of the domain of interest is inferred from the corpus in the form of a metagraph. Afterwards, once the domain structure has been discovered, named entity recognition (NER) and relation extraction (RE) models can be used to generate a domain-specific KG. We argue why the automated definition of the domain's structure as a first step is beneficial both in terms of construction time and quality of the generated graph. Furthermore, we present a machine learning approach, based on Transformers, to infer the structure of a corpus's domain. The proposed method is extensively validated on three public datasets (WebNLG, NYT and DocRED) by comparing it with two reference methods using CNNs and RNNs. Lastly, we demonstrate how this work lays the foundation for fully automated and unsupervised KG generation.

1 INTRODUCTION

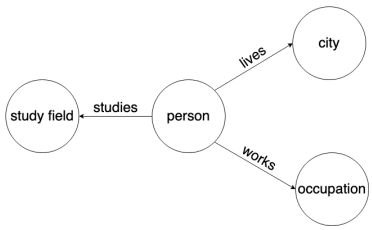
Knowledge Graphs (KGs) are among the most popular data management paradigms as they shares simultaneously several advantages of databases (information retrieval via structured queries), graphs (representing loosely or irregularly structured data) and knowledge bases (representing semantic relationship among the data). Therefore, they are ubiquitous in fields such as recommendation systems, question-answering tools and knowledge discovery applications. The continuously evolving KG research field (Ji et al., 2020) consists of two main areas: knowledge representation learning, which investigates the representation of KG into vector representations (KG embeddings), and knowledge acquisition, which considers the KG construction process. The latter being a fundamental aspect since a malformed graph will not be able to serve accurately any kind of future operations.

The KG construction phase can follow a bottom-up or top-down approach (Zhao et al., 2018). In a bottom-up approach, all the entities and their connections are extracted as a first step of the process. Then, the underlying hierarchy and structure of the domain can be inferred from the entities and their connections. On the other hand, a top-down approach starts with the definition of the domain's schema and then proceeds with the extraction of the needed entities and connections for the specific domain based on the underlying schema. For general KG construction, a bottom-up approach is usually preferred as we typically wish to include all the span of possible entities and relations that we can extract from the given corpus. Contrarily, a top-down approach suits better to a domain-specific KG construction, where entities and relations, and ineherently their extraction, are strongly linked to the domain of interest.

Our main interest is in domain-specific solutions for two main reasons. Firstly, focusing on a specific field, we incorporate into the graph only information that is relevant to the domain. This minimizes the presence of irrelevant data and restricts queries and graph operations to a carefully tailored KG. This generally improves the accuracy of KG applications (Lalithsena et al., 2016). Secondly, the graph's size is significantly reduced by excluding irrelevant content. Particularly, we can achieve a reduction of up 90% in specific use cases (Lalithsena et al., 2016) if we adhere to a domain-specific approach. Therefore the execution time of queries can be reduced by more than one order of magnitude.



(a) KG produced from the examples "John lives in London and studies computer science" and "Nick is a London based dentist".



(b) Domain's metagraph from the examples "John lives in London and studies computer science" and "Nick is a London based dentist".

Figure 1: Example of a KG and its respective domain's metragraph

The domain definition , here defined as a metagraph having entity types as nodes and relation types as edges, is usually performed by subject matter experts. Yet, KG construction by expert curation can be extremely slow as the process, in this case, is essentially manual. Moreover, human error may affect the data quality and lead to malformed KGs. In the context of this work, we propose to overcome these issues by introducing an automated machine learning-based approach to understand the domain of a given corpus. Specifically, we introduce a seq2seq-based model to infer the relation types characterizing the domain of interest. Equipped with this model, we can define the structure of the domain including all the needed entity and relation types that should be included in the graph in an automated manner and then utilize only the appropriate tools, such as specific entity and relation extractors, to populate the actual KG. We train such model using previous examples of text snippets and the respective relation types that are included in them. We show that our proposed model outperforms other baseline approaches and provide us with the needed high precision and recall combination for an accurate domain definition.

2 SEQ2SEQ-BASED MODEL FOR DOMAIN UNDERSTANDING

The domain understanding task attempts to uncover the structured knowledge underlying a dataset. In order to depict this structure we can leverage a so called domain's metagraph. A domain's metagraph is a graph that has as vertices all the entity types and as edges all their connections/relations in the context of this domain. To illustrate this definition, consider as a toy example that we have a dataset which contains only the following two sentences: "John lives in London and studies computer science" and "Nick is a London based dentist". Figure 1a depicts the KG that can be extracted from these two sentences and figure 1b the respective domain's metagraph.

The generation of such a metagraph requires obtaining all the entity types and their relations. Assuming that each entity type that is present in the domain has at least one interaction with another entity type, we can produce the metagraph of this domain by inferring all the possible entity type connections. Thus, our approach aims to build an accurate model of a domain's relation types, and leverages this model to extract the relation types from a given corpus. Aggregating all extracted relations yields the domain's metagraph.

2.1 SEQ2SEQ MODEL FOR DOMAIN'S RELATION EXTRACTION

Sequence to sequence models (seq2seq) (Bahdanau et al., 2014; Cho et al., 2014; Sutskever et al., 2014; Jozefowicz et al., 2016) attempt to learn the mapping from an input X to its corresponding target Y where both of them are represented by sequences: $X = \{x_1, x_2, .., x_s\}$ and $Y = \{y_1, y_2, ..., y_t\}$. They model such case using the conditional probability:

$$P(Y|X) = \prod_{t=1}^{n} P(y_t|y_1, y_2, ..., y_{t-1}, X)$$

Table 1: Examples of Normal, EntityPairOveral and SingleEntityOveralp text snippets

Type	Sentence	Triplets
`Normal`	London is the capital of UK	(London, capital_of, UK)
`EntityPairOverlap`	London is the capital of UK and its largest city	(London, capital_of, UK) (London, largest_city_of, UK)
`SingleEntityOverlap`	Heathrow airport is located at London which is the capital of UK	(Heathrow airport, located_at, London) (London, capital_of, UK)

To achieve this they follow an encoder-decoder based approach by having an encoder neural network to read sequentially each $x_s \in X$

$$h_s = encoder(h_{s-1}, x_s)$$

where h_s is the state of the encoder at time s and a decoder neural network to produce each $y_t \in Y$ given the current state g_t and the previous predicted symbol y_{t-1}

$$g_1 = h_s$$
$$g_t = decoder(g_{t-1}, y_{t-1})$$
$$P(y_t|y_1, y_2, ..., y_{t-1}, X) = softmax(g_t)$$

Encoder and decoders can be recurrent neural networks (Cho et al., 2014) or convolutional based neural networks (Gehring et al., 2017). In addition, an attention mechanism can also be incorporated into the encoder (Bahdanau et al., 2014; Luong et al., 2015) for further boosting of the model's performance. Lately, Transformer architectures (Vaswani et al., 2017; Devlin et al., 2018; Liu et al., 2019; Radford et al., 2018) , a family of models whose components are entirely made up of attention layers, linear layers and batch normalization layers, have established themselves as the state of the art for sequence modeling, outperforming the typically recurrent based components. Seq2seq models have been successfully utilized for various tasks such as neural machine translation (Bahdanau et al., 2014) and natural language generation (Pust et al., 2015) and their scope has been extended beyond language processing in fields such as chemical reaction prediction (Schwaller et al., 2019).

We consider the domain's relation type extraction task as a specific version of machine translation from the language of the corpus to the "relation" language that includes all the different relations between the entity types of the domain. A relation type R which connects the entity type i to j is represented as "i.R.j" in the "relation" language. In the case of undirected connections, "i.R.j" is the same as "j.R.i" and for simplicity we can discard one of them.

Seq2seq models have been designed to address tasks where both the input and the output sequences are ordered. In our case the target "relation" language does not have any defined ordering as per definition the edges of a graph do not have any ordering. In theory the order does not matter, yet in practice unordered sequences will lead to slower convergence of the model and requirements of more training data to achieve our goal (Vinyals et al., 2015). To overcome this issue, we propose a specific ordering of the "relation" language influenced from the semantic context that the majority of the text snippets hold.

According to Zeng et al. (2018), for the standard relation extraction task, text snippets can be divided into three types: `Normal`, `EntityPairOverlap` and `SingleEntityOverlap`. A text snippet is categorized as `Normal` if none of its triplets have overlapping entities. If some of its triplets express a relation on the same pair of entities then it belongs to the `EntityPairOverlap` category and if some of its triplets have one entity in common but no overlapped pairs, then it belongs to `SingleEntityOverlap` class. Table 1 presents one example for each category. These three categories are also relevant in the metagraph case, even if we are working with entity types and relation types rather than the actual entities and their relations.

Based on the given training set, we consider that the model is aware of the general domain anatomy, i.e., all the entity types and potential relations are known, and we would like to identify which of them are depicted in a given corpus. In both cases of `EntityPairOverlap` and

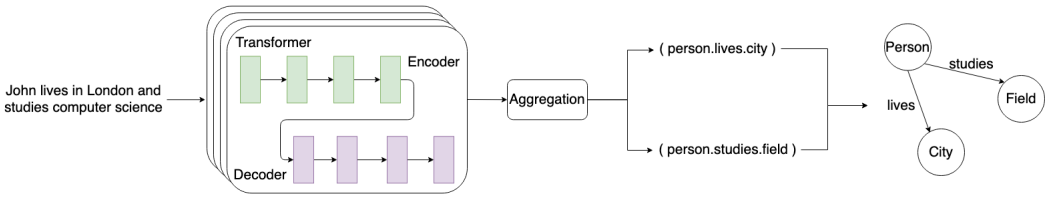


Figure 2: Architecture of our utilized Transformer model.

`SingleEntityOverlap` type text snippets there is one main entity type from which all the other entity types can be found by performing only one hop traversal in the general domain's metagraph. The class of `Normal` text snippets is a more broad case in which one can identify heterogeneous connectivity patterns among the entity types that are described in them. Yet, a sentence typically describes facts that are expected to be connected somehow, thus the entity types included in such texts usually are not more than 1 or 2 hops away from each other in the general metagraph. Considering the above, we propose to sort the relations in a breadth-first-search (BFS) order starting from a specific node (entity type) in the general metagraph. In this way, we confine the output in a much lower dimensional space by adhering to a semantically meaningful order.

Inspired by state-of-the-art approaches in the field of neural network translation, our model architecture is a multi-layer bidirectional Transformer. We follow the lead of Vaswani et al. (2017) in implementing the architecture, with the only difference that we adopt a learned positional encoding instead of a static one (see A.1 for further details for the positional encoding). As the overall architecture of the encoder and the decoder are otherwise the same as in Vaswani et al. (2017), we omit an in-depth description of these components and refer readers to original paper.

To boost the model's performance, we also propose an ensemble approach by utilizing several different Transformers and aggregating their results to construct the domain's metagraph. The differentiation point from each Transformer is the selected ordering of the "relation" vocabulary. The selection of different starting entity type for the breadth-first-search will lead to different ordering. We expect that multiple orderings could facilitate the prediction of different connection patterns which recognition based solely in one ordering may not be feasible. The sequence of steps for an ensemble domain understanding is the following: Firstly, train k Transformers using different ordering for each of them. Secondly, given a set of text snippets, predict sequences of relations using all the Transformers. Finally, utilize an ensemble method to aggregate the results and form the final predictions.

It is worth mentioning that in the last step, we omit the underling ordering that we follow in each model and we perform a relation-based aggregation. We examine each relation separately in order to include it or not in the final metagraph. For the aggregation step, we use the standard Wisdom of Crowds (WOC) (Marbach et al., 2012) consensus technique, yet other consensus methods can also be leveraged for the task. The overall structure of our architecture is presented in Figure 2.

3 Related work

At the best of our knowledge, our method is the first attempt to introduce a domain understanding component in the process of KG generation. The two closest explored research areas are the relation extraction and the Ontology learning fields. The former attempts to extract semantic relations for texts while the latter aims at generate ontologies that describe the concepts and their relations in a domain.

The relation extraction task aims at the extraction of triplets of the form of (subject, relation, object) from the texts. The neural network based methods, such as Nguyen & Grishman (2015); Zhou et al. (2016); Zhang et al. (2017), dominate the field. These methods are CNN (Zeng et al., 2014; Nguyen & Grishman, 2015) or LSTM (Zhou et al., 2016; Zhang et al., 2017) models which given a text and information about the position of entities in it attempt to identify their relations. The positional information of the entities is typically extracted in a previous step of KG generation using named entity recognition (NER) methods (Nadeau & Sekine, 2007). Lately, there is a high interest

Table 2: Datasets's statistics

Dataset	# instances	# entity types	# relation types	size of "relation" language
WebNLG	23794	45	47	70
NYT	70029	12	28	31
DocRED	30289	6	96	511

of methods that can combine the NER and relation extractions tasks into a single model (Zheng et al., 2017; Zeng et al., 2018; Fu et al., 2019). Our task is related to the relation extraction task as we also interested in extraction of relation types from a text. Nevertheless, as we place our task in the beginning of the KG generation process, we do not have available any information about the position of the entities in the text neither we attempt to identify them. We solely focus on the relation types and the acquisition of the domain's metagraph in order to utilize the proper models in the next steps of the pipeline, which also means to utilize the proper NER or relation extraction models for the domain of interest.

Ontology learning aims at representing the knowledge of a domain by providing all of its concepts and their relations. Currently, the proposed pipelines (Drymonas et al., 2010; Venu et al., 2016) attempt to adopt end-to-end methods for ontology learning. These pipelines are based on pattern and association rule mining and they perform sequentially term extraction, relation extraction and lastly the ontology building. The automated ontology construction is particularly used for Intelligent Tutoring Systems (ITS) where they try to improve the learning process by representing the domain of a subject to automatically find the best learning path for the students. The main interest for ITS are taxonomic relations like "is-a" and "part-of". In Larranaga et al. (2013) they propose a method to define domains describing in textbooks. They analyze both the outline and the document body using mainly heuristic and rule-based techniques to extract relations between concepts. Recently, deep learning has been also utilized in the field (Navarro-Almanza et al., 2020). Specifically, they utilize a Bidirectional Gate Recurrent Neural Network (RNN) with attention model for the relation type detection task. Furthermore, they propose a transfer learning technique to adapt the model into different domains. This approach is the first to connect the ontology learning and relation extraction fields as it leverages the model that has been described in Zhou et al. (2016) for the relation extraction. Our method is highly correlated with the ontology learning task as we also want to represent the domain of interest. Therefore, we rely solely on deep learning techniques, something not fully explored in the field yet. Last, we are not confined to identification of taxonomic relations only, which is the case for many applications of Ontology learning such as the ITS systems.

4 EXPERIMENTS

We evaluate our Transformer-based approach against three baselines on a selection of datasets representing different domains. As baselines, we use the CNN and the RNN based relation type extraction methods that is used in Nguyen & Grishman (2015) and Zhou et al. (2016) respectively. For both methods, we slightly modified the architectures to exclude the component which provides information about the position of the entities in the text snippet, as we do not have such information available in our task. Additionally, we also include a Transformer-based model without applying any ordering in the target sequences as an extra baseline. We compare based on accuracy and F1-score. Accuracy is computed at an instance level as we examine how many target sentences are correct over all the testing set. The ordering of the target sentence does not assessed during the evaluation as we only examine the existence of each relation in the target and not its position. F1-score is the harmonic mean of the precision and recall and it is computed at a relation level. For our proposed method, we present the BFS based ordering and the ensemble variant which have the best accuracy.

To our knowledge, there is no standard dataset available for the relation type extraction task in the literature, however there is a plethora of published datasets for the standard task of relation extraction that can be utilized for the relation type extraction task with limited effort. We use WebNLG (Gardent et al., 2017), NYT (Riedel et al., 2010) and DocRED (Yao et al., 2019), three of the most popular datasets for relation extraction. All these three datasets contain instances with more than one relation in it. Both NYT and DocRED datasets provide the needed information such as entity types and relation type for each of their instances, so their transformation for our specific task is

Table 3: Comparison of CNN model, RNN model and Transformer-based methods on WebNLG, NYT and DocRED datasets. *The architecture of the CNN and RNN models has been modified to exclude the component which provides information about the position of the entities in the text snippet.

Dataset	Model	Accuracy	F1 score
WebNLG	CNN (Nguyen & Grishman, 2015)*	0.8156 ± 0.0071	0.9459 ± 0.0021
	RNN (Zhou et al., 2016)*	0.8517 ±0.0058	0.9543 ± 0.0021
	Transformer - unordered	**0.8798 ±0.0053**	**0.9646 ± 0.0018**
	Transformer - BFS_{record_label}	**0.9000 ± 0.0046**	**0.9699 ± 0.0013**
	Transformer - WOC k=20	**0.9235 ± 0.0014**	**0.9780 ± 0.0003**
NYT	CNN (Nguyen & Grishman, 2015)*	0.7341 ± 0.0035	**0.8385 ± 0.0025**
	RNN (Zhou et al., 2016)*	**0.7520 ± 0.0027**	**0.8353 ± 0.0029**
	Transformer - unordered	0.7426 ± 0.0061	0.8009 ± 0.0057
	Transformer - BFS_{person}	**0.7491 ± 0.0048**	0.8049 ± 0.0073
	Transformer - WOC k=8	**0.7669 ± 0.0011**	**0.8307 ± 0.0006**
DocRED	CNN (Nguyen & Grishman, 2015)*	0.1096 ± 0.0073	0.4434 ± 0.0133
	RNN (Zhou et al., 2016)*	0.2178± 0.0088	0.6192 ± 0.0093
	Transformer - unordered	**0.4869 ±0.0069**	**0.7081 ±0.0032**
	Transformer - BFS_{ORG}	**0.5252 ± 0.0048**	**0.7133 ± 0.0049**
	Transformer - WOC k=6	**0.5722 ± 0.0001**	**0.7607 ± 0.0001**

trivial. On the other hand, WebNLG doesn't share such information for the entity types and we performed a manual transformation by examining all the possible entities that a relation connects and replace them with the proper entity type. For the WebNLG dataset, we avoid to include rare entity and relation types and we either omit them or replace them with similar or more general types that exists in it (see A.2 for further details). Table 2 depicts the statistics of the three datasets.

For all datasets, we use the same model parameters. Specifically, we use Adam (Kingma & Ba, 2014) optimizer with a learning rate of 0.0005. The gradients norm is clipped to 1.0 and dropout (LeCun et al., 2015) is set to 0.1. Both encoder and decoder consist of 3 layers with 10 attention heads each, the positional feed-forward hidden dimension is 512. Lastly, we utilize the token embedding layers using GloVe pretrained word embeddings (Pennington et al., 2014) which have dimensionality of m=300. Our code has been anonymously available at `https://anonymous.4open.science/r/170261fd-ba97-468b-a32f-e9d72b763747`

INSTANCE LEVEL EVALUATION OF THE MODELS

To study the performance of our model, we perform 10 independent runs each with different random splitting of the datasets into training, validation and testing set. Table 3 depicts the median value and the standard error of the baselines and our method for the two metrics. Our method is better in terms of accuracy for all the three datasets and in terms of F1-score for the WebNLG and DocRED datasets. For the NYT dataset, the F1-score of CNN and RNN models outperform our approach. We observed that the baseline models profit from the fact that, in this dataset, the majority of the instances depict only one relation and many of the relations appear in a limited number of instances. In general, there is lack of sequences of relations that hinders the Transformer's ability to learn the underlying distribution(see A.3). Lastly, the decreased performances of all the models in the DocRED dataset is due to the long tail characteristic that this dataset shows as 66% of the relations appeared in no more than 50 instances (see A.4).

GRAPH LEVEL EVALUATION OF THE MODELS

The above comparisons only focus on the ability of the model to predict the relation types given a corpus. Since our ultimate goal is to infer the domain's metagraph for each dataset, we generate 10 subdomains by selecting randomly 10 instances from each of the testing sets. We infer the relations types for each instance and then we generate the domain's metagraph by simply including all the relation types that were found in the instances. Then, we compare how close the actual domain's metagraph and the predicted metagraph are. We examine the F1-score for both edges and nodes

Table 4: Evaluation of metagraph's reconstruction on the three datasets using CNN, RNN and Transformer-based models. *The architecture of the CNN and RNN models has been modified to exclude the component which provides information about the position of the entities in the text snippet.

Dataset	Model	Edges F1-score	Nodes F1-score	Degree JSD	Eigenvector JSD
WebNLG	CNN (Nguyen & Grishman, 2015)*	**0.9747**	**0.9879**	**0.1836**	**0.2059**
	RNN (Zhou et al., 2016)*	0.9639	0.9735	0.2708	0.2364
	Transformer - unordered	0.9598	**0.9775**	0.2380	0.2280
	Transformer - BFS_{record_label}	**0.9806**	**0.9772**	**0.1923**	**0.1593**
	Transformer - WOC k=5	**0.9808**	**0.9772**	**0.1765**	**0.1261**
NYT	CNN (Nguyen & Grishman, 2015)*	**0.9059**	**0.9800**	0.0564	0.0832
	RNN (Zhou et al., 2016)*	**0.9205**	1	**0**	**0**
	Transformer - unordered	0.8184	**0.9800**	0.0967	0.1396
	Transformer - BFS_{person}	**0.8806**	0.9666	**0**	**0**
	Transformer - WOC k=8	0.8672	**1**	**0**	**0**
DocRED	CNN (Nguyen & Grishman, 2015)*	0.4819	0.9019	0.5717	0.6965
	RNN (Zhou et al., 2016)*	0.6823	0.9714	0.5187	0.6954
	Transformer - unordered	**0.7530**	**1**	**0.2950**	**0.5997**
	Transformer - BFS_{PER}	**0.7830**	**0.9777**	**0.2892**	**0.4267**
	Transformer - WOC k=6	**0.8045**	**1**	**0.2349**	**0.3688**

of the predicted graph as well as the similarity of the distribution of the degree and eigenvector centrality (Zaki & Meira, 2014) of the two metagraphs. For the comparison of the centralities distribution, we construct the histogram of the centralities for each graph using 10 fixed size bins and we utilize Jensen-Shannon Divergence (JSD) metric (Endres & Schindelin, 2003) to examine the similarity of the two distributions (see A.7 for the definition of JSD). We have selected degree and eigenvector centralities as the former gives as localized structure information as measure the importance of a node based on the direct connections of it and the latter gives as a broader structure information as measure the importance of a node based on infinite walks.

Table 4 presents the results of the evaluation of the predicted versus the actual domain's metagraph for the 10 subdomains extracted from the the testing set of the WebNLG dataset. All the presented values are the mean over all the 10 subdomains. Our approach using Transformer + BFS based ordering outperforms or is close to the baselines for all cases in terms of edges and nodes F1-score. Furthermore, the degree and eigenvector centralities distribution of the generated metagraphs using our method are closer to the groundtruth in comparison to other methods in all cases. This indicates that the graphs produced with our method are both element-wise and structurally closer to the actual ones.

More detailed comparisons of the different methods at both instance and metagraph level have been included in the Appendix (see A.5).

The ensemble variant of our approach based on the WOC consensus strategy outperforms the simple Transformer + BFS ordering in all cases. Based on the evaluation at both instance and metagraph level, our ensemble variant seems to be the most reliable approach for the task of domain's relation type extraction. In the Appendix (see A.6), we have also included results of a second consensus strategy, where adopting a user-defined cut-off we can indicate whether the focus on a high precision or on a high recall outcome.

TOWARDS AUTOMATED KG GENERATION

The proposed domain understanding method enables the inference of the domain of interest and its components. This enables a partial automation and a speed up of the KG generation process as, without manual intervention, we are able to identify the metagraph, and inherently the needed models for the entity and relation extraction in the context of the domain of interest. To achieve this, we adopt a Transformer-based approach that heavily relies on attention mechanisms. Recent efforts are focusing on the analysis of such attention mechanisms to explain and interpret the predictions and the quality of the models (Vig & Belinkov, 2019; Hoover et al., 2019). Interestingly, it has been

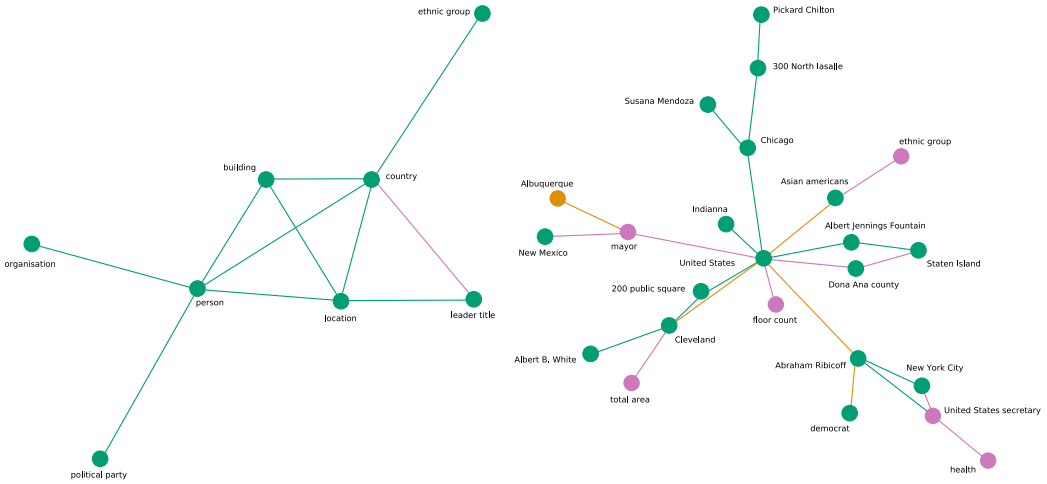


Figure 3: Metagraph (left) and the KG (right) extracted from 12 text snippets related to the United States using our model and the respective attention analysis. The colors in the nodes/edges mean the following: green exists in both actual and predicted graphs, orange exists in the actual but not in the predicted graph, pink exists in the predicted but not in the actual.

shown how the analysis of the attention pattern can elucidate complex relations between the entities fed as input to the Transformer, e.g., mapping atoms in chemical reactions with no supervision (Schwaller et al., 2020). Even if it is out of the scope of our current work, we observe that a similar analysis of the attention patterns in our model can identify not only parts of text in which relations exist but directly the entities of the respective triplets. To illustrate this, we extract 12 text instances from the WebNLG and after generating the domain's metagraph, we analyze the attention to triples to build a KG. We rely on the syntax dependencies to propagate the attention weights throughout the connected tokens and we examine the noun chunks to extract the entities of interest based on their accumulated attention weight (see A.8 for further details). We select the head which achieves the best precision and recall in order to generate the KG. Figure 3 depicts the generated metagraph and the KG. Using the aforementioned attention analysis, we manage to achieve 0.76 precision and 0.95 recall in the entity extraction and 0.57 precision and 0.70 recall in the relation extraction. These values might not be able to compete the state of the art respective models and the investigation is limited in only few instances. Yet it indicates that a completely unsupervised generation based on attention analysis is possible and deserves further investigation.

5 Conclusion

Herein, we proposed a method to speed up the KG generation task by defining the domain of interest in an automated manner. This is achieved by using a Transformer-based approach to estimate the metagraph representing the schema of the domain. The evaluation and the comparison over different datasets against state-of-the-art methods indicates that our approach accurately produces the metagraph. This paves the way towards an automated KG generation, where after predicting the metagraph of the domain, the construction process reduces to the selection of appropriate named entity recognition and relation extraction models. We believe that the method proposed is key to minimize the need of human intervention in the KG construction process, hence allowing in the near future to avoid the currently needed manual curation. It is also important to notice that, as a side effect, such attention-based model can be directly applied to triplet extraction from the text without retraining and without supervision. Triplet extraction in an unsupervised way represents a breakthrough, especially if combined with most recent advances in zero-shot learning for NER (Li et al., 2020; Pasupat & Liang, 2014; Guerini et al., 2018), and we believe, relying on the preliminary studies conducted, that our Transformer-based approach should be investigated further in this direction.

REFERENCES

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Euthymios Drymonas, Kalliopi Zervanou, and Euripides GM Petrakis. Unsupervised ontology acquisition from plain texts: the ontogain system. In *International Conference on Application of Natural Language to Information Systems*, pp. 277–287. Springer, 2010.

Dominik Maria Endres and Johannes E Schindelin. A new metric for probability distributions. *IEEE Transactions on Information theory*, 49(7):1858–1860, 2003.

Tsu-Jui Fu, Peng-Hsuan Li, and Wei-Yun Ma. Graphrel: Modeling text as relational graphs for joint entity and relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1409–1418, 2019.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. Creating training corpora for nlg micro-planning. In *55th annual meeting of the Association for Computational Linguistics (ACL)*, 2017.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1243–1252. JMLR. org, 2017.

Marco Guerini, Simone Magnolini, Vevake Balaraman, and Bernardo Magnini. Toward zero-shot entity recognition in task-oriented conversational agents. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pp. 317–326, 2018.

Benjamin Hoover, Hendrik Strobelt, and Sebastian Gehrmann. exbert: A visual analysis tool to explore learned representations in transformers models. *arXiv preprint arXiv:1910.05276*, 2019.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S Yu. A survey on knowledge graphs: Representation, acquisition and applications. *arXiv preprint arXiv:2002.00388*, 2020.

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*, 2016.

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

Sarasi Lalithsena, Pavan Kapanipathi, and Amit Sheth. Harnessing relationships for domain-specific subgraph extraction: A recommendation use case. In *2016 IEEE International Conference on Big Data (Big Data)*, pp. 706–715. IEEE, 2016.

Mikel Larranaga, Angel Conde, Inaki Calvo, Jon A Elorriaga, and Ana Arruarte. Automatic generation of the domain module from electronic textbooks: method and validation. *IEEE transactions on knowledge and data engineering*, 26(1):69–82, 2013.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

Daniel Marbach, James C Costello, Robert Küffner, Nicole M Vega, Robert J Prill, Diogo M Camacho, Kyle R Allison, Manolis Kellis, James J Collins, and Gustavo Stolovitzky. Wisdom of crowds for robust gene network inference. *Nature methods*, 9(8):796–804, 2012.

David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

Raúl Navarro-Almanza, Reyes Juárez-Ramírez, Guillermo Licea, and Juan R Castro. Automated ontology extraction from unstructured texts using deep learning. In *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, pp. 727–755. Springer, 2020.

Thien Huu Nguyen and Ralph Grishman. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pp. 39–48, 2015.

Panupong Pasupat and Percy Liang. Zero-shot entity extraction from web pages. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 391–401, 2014.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.

Michael Pust, Ulf Hermjakob, Kevin Knight, Daniel Marcu, and Jonathan May. Parsing english into abstract meaning representation using syntax-based machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1143–1154, 2015.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training, 2018.

Sebastian Riedel, Limin Yao, and Andrew McCallum. Modeling relations and their mentions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 148–163. Springer, 2010.

Philippe Schwaller, Teodoro Laino, Théophile Gaudin, Peter Bolgar, Christopher A. Hunter, Costas Bekas, and Alpha A. Lee. Molecular transformer: A model for uncertainty-calibrated chemical reaction prediction. *ACS Central Science*, 5(9):1572–1583, 2019.

Philippe Schwaller, Benjamin Hoover, Jean-Louis Reymond, Hendrik Strobelt, and Teodoro Laino. Unsupervised attention-guided atom-mapping, May 2020.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pp. 3104–3112, 2014.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Sree Harissh Venu, Vignesh Mohan, Kodaikkaavirinaadan Urkalan, and TV Geetha. Unsupervised domain ontology learning from text. In *International Conference on Mining Intelligence and Knowledge Exploration*, pp. 132–143. Springer, 2016.

Jesse Vig and Yonatan Belinkov. Analyzing the structure of attention in a transformer language model. *arXiv preprint arXiv:1906.04284*, 2019.

Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. Order matters: Sequence to sequence for sets. *arXiv preprint arXiv:1511.06391*, 2015.

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. Docred: A large-scale document-level relation extraction dataset. *arXiv preprint arXiv:1906.06127*, 2019.

Mohammed J Zaki and Wagner Meira. *Data mining and analysis: fundamental concepts and algorithms*. Cambridge University Press, 2014.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pp. 2335–2344, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics.

Xiangrong Zeng, Daojian Zeng, Shizhu He, Kang Liu, and Jun Zhao. Extracting relational facts by an end-to-end neural model with copy mechanism. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 506–514, 2018.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 35–45, 2017.

Zhanfang Zhao, Sung-Kook Han, and In-Mi So. Architecture of knowledge graph construction techniques. *International Journal of Pure and Applied Mathematics*, 118(19):1869–1883, 2018.

Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. Joint extraction of entities and relations based on a novel tagging scheme. *arXiv preprint arXiv:1706.05075*, 2017.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 207–212, 2016.

A Appendix

A.1 Learned positional encoding

In our model, we adopt a learned positional encoding instead of a static one. Specifically, the tokens are passed through a standard embedding layer as a first step in the encoder. The model has no recurrent layers and therefore it has no idea about the order of the tokens within the sequence. To overcome this, we utilize a second embedding layer called a positional embedding layer. This is a standard embedding layer where the input is not the token itself but the position of the token within the sequence, starting with the first token, the <sos> (start of sequence) token, in position 0. The position embedding has a "vocabulary" size equal to the maximum length of the input sequence. The token embedding and positional embedding are element-wise summed together to get the final token embedding which contains information about both the token and its position within the sequence. This final token embedding is then provided as input in the stack of attention layers of the encoder.

A.2 WebNLG dataset transformation

The transformation of the WebNLG dataset, in order to be used in the relation type extraction task, was done manually based on the following criteria. We inspect the subject and the object of each relation type and we replace the entities with the respective entity types. We try to produce a dataset where all the entity and relation types are included in a reasonable number of instances. For this reason, similar entity types or relation types with few instances (less than 10) have been merged together to form more general semantic concepts. For example, the entity types `city`, `county`, `area` have been merged together to form the entity type `location` and the relation types that indicates connection between a person and a location such as (place of birth, place of death, etc) have been consolidated into the relation type `location.related_to.person`. Finally, triplets that include rare concepts (concepts which exist in less than 5 instances) that cannot be merged with other entity/relation types have been excluded.

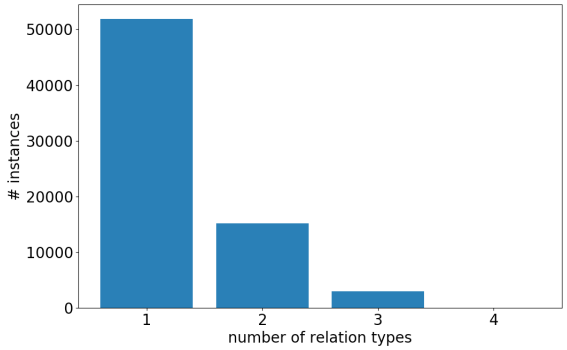


Figure 4: Histogram that depicts the number of instances that hold the respective number of relation types the for NYT dataset

Table 5: Performance of CNN, RNN and Transformer-based methods on instances from NYT datasets with more than 1 relation in them. *The architecture of the CNN and RNN models has been modified to exclude the component which provides information about the position of the entities in the text snippet.

Model	Accuracy	Precision	Recall	F1-score
CNN (Nguyen & Grishman, 2015)*	0.6487 ± 0.0183	**0.9549 ± 0.0039**	0.7751 ± 0.0131	0.8558 ± 0.0087
RNN (Zhou et al., 2016)*	0.6941 ± 0.0110	**0.9345 ± 0.0037**	0.8043 ± 0.0076	0.8645 ± 0.0051
Transformer - unordered	0.7035 ± 0.0253	0.9104 ± 0.0061	0.7938 ± 0.0161	0.8481± 0.0115
Transformer - $BFS_{location}$	0.7112 ± 0.0250	0.9168 ± 0.0062	0.8027 ± 0.0193	0.8559 ± 0.0129
Transformer - BFS_{person}	0.7097 ± 0.0146	0.9128 ± 0.0043	0.8016 ± 0.0095	0.8536 ± 0.0062
Transformer - $BFS_{company}$	0.7084 ± 0.0210	0.9148 ± 0.0049	0.7983 ± 0.0129	0.8525 ± 0.0091
Transformer - WOC k=4	**0.7379 ± 0.0055**	0.9207 ± 0.0037	**0.8321 ± 0.0052**	**0.8741 ± 0.0035**
Transformer - WOC k=8	**0.7430 ± 0.0029**	0.9303 ± 0.0016	**0.8269 ± 0.0021**	**0.8755 ± 0.0011**
Transformer - WOC k=12	**0.7433 ± 0.0014**	**0.9332 ± 0.0004**	**0.8250 ± 0.0011**	**0.8757 ± 0.0005**

A.3 Evaluation of the models in NYT dataset's instances with more than one relation

In the main text, we have indicated that the poor performance of our model in NYT dataset is due to the fact that the majority of its instances have only one relation. This lack of sequences of relations hinders the Transformer's ability to learn the underlying distribution. Thus, CNN and RNN based models manage to perform better than our approach there. To justify this, we first present in figure 4 the histogram of the relation types that the instances of the dataset hold. As it can be extracted from the figure the 75% of the dataset is one-relation type instances. Secondly, we utilize our model and the baselines in the instances of the NYT testing sets with more than one relation. Table 5 depicts the results. From the table and especially from the accuracy and F1-score, we can justify that our approach performs better than the baselines in the cases where the actual output is sequence.

A.4 DocRED dataset characteristics

We attributed the decreased performances of all the models in the DocRED dataset in its long tail characteristic that it holds. To justify this, figure 5 depicts the number of appearances for all the relation types of the dataset. Almost the 50% of the relations appeared in no more than 10 instances and the 66% of the relations appeared in no more than 50 instances.

A.5 Models's comparison

Table 6 presents a detailed evaluation of our approach and the baselines models. We have included the 3 best BFS ordering variants (in terms of accuracy) and 3 consensus variants. To cover the range of all the available values of k, [1, number of entity types], we select a case with just a few Transformers, one with a value close to half of the total number of entity types and one close to the total number of entity types. For each different k value, we utilize the top k best orderings based on

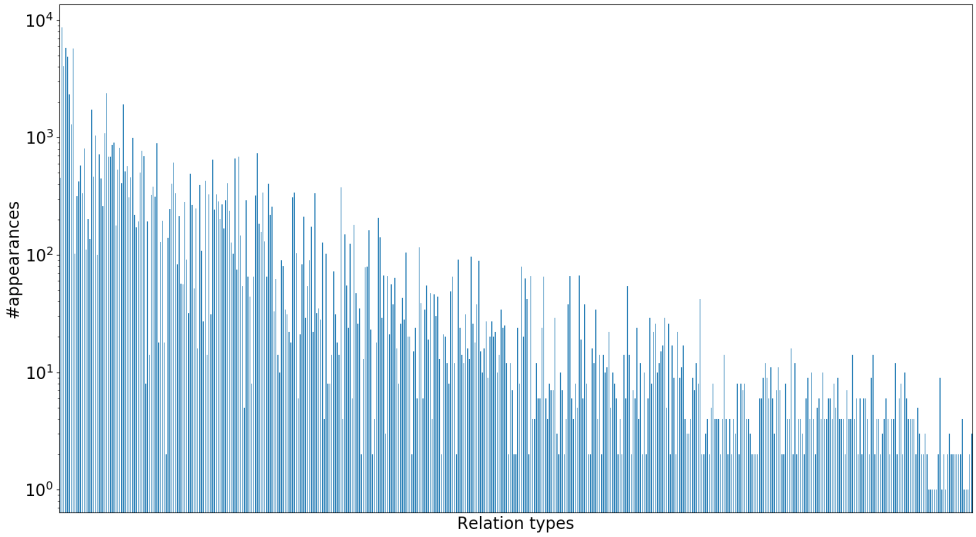


Figure 5: Number of appearances of the relation types in the DocRED dataset

Table 6: Comparison of CNN, RNN and Transformer-based methods on WebNLG, NYT and DocRED datasets for the relation type extractiont task. *The architecture of the CNN and RNN models has been modified to exclude the component which provides information about the position of the entities in the text snippet.

Dataset	Model	Accuracy	Precision	Recall	F1 score
WebNLG	CNN (Nguyen & Grishman, 2015)*	0.8156 ± 0.0071	0.9550 ± 0.0029	0.9370 ± 0.0049	0.9459 ± 0.0021
	RNN (Zhou et al., 2016)*	0.8517 ±0.0058	0.9614 ±0.0022	0.9472±0.0043	0.9543 ±0.0021
	Transformer - unordered	0.8798 ±0.0053	0.9678 ±0.0032	0.9614 ±0.0042	0.9646 ±0.0018
	Transformer - $BFS_{occupation}$	0.8987±0.0068	0.9693±0.0030	0.9705 ± 0.0035	0.9699 ± 0.0018
	Transformer - BFS_{music_genre}	0.8983 ± 0.0053	0.9694 ±0.0039	0.9703 ±0.0030	0.9699 ± 0.0015
	Transformer - BFS_{record_label}	0.9000 ± 0.0046	0.9707 ± 0.0031	0.9691 ±0.0028	0.9699 ±0.0013
	Transformer - WOC k=5	**0.9210 ± 0.0017**	**0.9789 ±0.0016**	**0.9755 ± 0.0013**	**0.9772 ± 0.0004**
	Transformer - WOC k=20	**0.9235 ± 0.0014**	**0.9786 ±0.0006**	**0.9774 ± 0.0004**	**0.9780 ±0.0003**
	Transformer - WOC k=45	**0.9235 ± 0.0002**	**0.9795 ±0.0001**	**0.9767 ±0.0001**	**0.9781 ±0.0001**
NYT	CNN (Nguyen & Grishman, 2015)*	0.7341 ± 0.0035	**0.8873 ± 0.0032**	0.7948 ± 0.0053	**0.8385 ± 0.0025**
	RNN (Zhou et al., 2016)*	0.7520 ± 0.0027	**0.8530 ±0.0041**	0.8183 ±0.0047	**0.8353 ±0.0029**
	Transformer - unordered	0.7426 ± 0.0061	0.8059 ±0.0079	0.7960 ± 0.0070	0.8009 ±0.0057
	Transformer - $BFS_{location}$	0.7461 ± 0.0053	0.8093 ±0.0067	0.7988 ±0.0067	0.8040 ±0.0043
	Transformer - BFS_{person}	0.7491 ±0.0048	0.8119 ± 0.0036	0.8022 ±0.0032	0.8049 ± 0.0073
	Transformer - $BFS_{company}$	0.7461 ±0.0081	0.8056 ±0.0088	0.8043 ±0.0117	0.8049 ±0.0073
	Transformer - WOC k=4	**0.7547 ±0.0038**	0.8162 ± 0.0047	**0.8355 ± 0.0022**	0.8257 ±0.0023
	Transformer - WOC k=8	**0.7669 ±0.0011**	**0.8325 ± 0.0025**	**0.8289 ±0.0018**	0.8307 ± 0.0006
	Transformer - WOC k=12	**0.7698 ± 0.0007**	0.8381 ±0.0017	**0.8296 ±0.0009**	**0.8320 ±0.0004**
DocRED	CNN (Nguyen & Grishman, 2015)*	0.1096 ± 0.0073	0.7838 ± 0.0081	0.3094 ± 0.0140	0.4434 ± 0.0133
	RNN (Zhou et al., 2016)*	0.2178± 0.0088	0.7716 ±0.0100	0.5173 ±0.0143	0.6192 ± 0.0093
	Transformer - unordered	0.4869 ±0.0069	0.7365 ±0.0156	0.6822 ±0.0127	0.7081 ±0.0032
	Transformer - BFS_{LOC}	0.5235 ± 0.0049	0.7145 ± 0.0112	0.7053 ±0.0076	0.7098 ± 0.0040
	Transformer - BFS_{PER}	0.5234 ± 0.0077	0.7234 ± 0.0179	0.7029 ± 0.0091	0.7128 ±0.0060
	Transformer - BFS_{ORG}	0.5252 ± 0.0048	0.7216 ± 0.0068	0.7053 ± 0.0069	0.7133 ± 0.0049
	Transfomer - WOC k=4	**0.5678 ±0.0037**	**0.7939 ±0.0137**	**0.7227 ±0.0103**	**0.7564 ± 0.0032**
	Transformer - WOC k=5	**0.5697 ± 0.0016**	**0.8012 ±0.0112**	**0.7210 ±0.0053**	**0.7589 ±0.0025**
	Transformer - WOC k=6	**0.5722 ± 0.0001**	**0.7970 ± 0.0035**	**0.7276 ± 0.0022**	**0.7607 ± 0.0001**

their accuracy. In addition to the per instance accuracy and the per relation F1-score, the table also includes the per relation precision and the recall of each model. Our proposed method, especially its ensemble variant, produces the best outcome in all datasets apart from the NYT case where the CNN and RNN models manage to be more precise. This is attributed to the characteristics of NYT dataset, where there are many one-only relation instances.

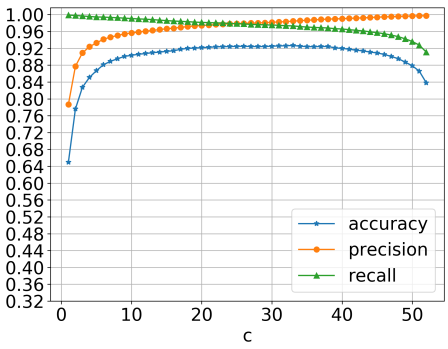
Similarly, in table 7 we perform an in-depth graph level evaluation of the models. For all three datasets, our proposed method and its ensemble extension produce the best or one of the top-3 best outcomes.

Table 7: Evaluation of metagraph's reconstruction on WebNLG dataset using CNN, RNN and Transformer-based models. *The architecture of the CNN and RNN models has been modified to exclude the component which provides information about the position of the entities in the text snippet.

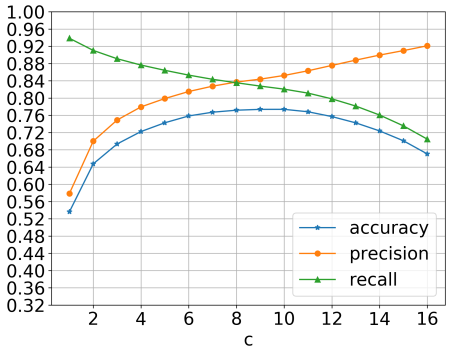
Dataset	Model	Edges F1-score	Nodes F1-score	Degree JSD	Eigenvector JSD
WebNLG	CNN (Nguyen & Grishman, 2015)*	0.9747	**0.9879**	0.1836	0.2059
	RNN (Zhou et al., 2016)*	0.9639	0.9735	0.2708	0.2364
	Transformer - unordered	0.9598	0.9775	0.2380	0.2280
	Transformer - $BFS_{occupation}$	**0.9831**	0.9805	0.1743	0.1840
	Transformer - BFS_{music_genre}	0.9755	0.9746	**0.1131**	0.1306
	Transformer - BFS_{record_label}	0.9806	0.9772	0.1923	0.1593
	Transformer - WOC k=5	0.9808	0.9772	0.1765	**0.1261**
	Transformer - WOC k=20	**0.9864**	**0.9840**	**0.1456**	**0.070**
	Transformer - WOC k=45	**0.9930**	**0.9916**	**0.1313**	**0.0891**
NYT	CNN (Nguyen & Grishman, 2015)*	**0.9059**	0.9800	0.0564	0.0832
	RNN (Zhou et al., 2016)*	**0.9205**	**1**	**0**	**0**
	Transformer - unordered	0.8184	0.9800	0.0967	0.1396
	Transformer - $BFS_{location}$	0.8141	0.9800	0.0832	0.0832
	Transformer - BFS_{person}	**0.8806**	0.9666	**0**	**0**
	Transformer - $BFS_{company}$	0.8305	0.9657	**0**	**0**
	Transformer - WOC k=4	0.8442	**1**	**0**	**0**
	Transformer - WOC k=8	0.8672	**1**	**0**	**0**
	Transformer - WOC k=12	0.8666	**1**	**0**	**0**
DocRED	CNN (Nguyen & Grishman, 2015)*	0.4819	0.9019	0.5717	0.6965
	RNN (Zhou et al., 2016)*	0.6823	0.9714	0.5187	0.6954
	Transformer - unordered	0.7530	**1**	0.2950	0.5997
	Transformer - BFS_{LOC}	0.7710	**1**	0.2744	0.5140
	Transformer - BFS_{PER}	0.7830	0.9777	0.2892	**0.4267**
	Transformer - BFS_{ORG}	0.7243	0.9777	0.2892	**0.4267**
	Transformer - WOC k=4	**0.7997**	**1**	**0.2714**	0.4787
	Transformer - WOC k=5	**0.8090**	**1**	**0.2673**	0.4433
	Transformer - WOC k=6	**0.8045**	**1**	**0.2349**	**0.3688**

A.6 ENSEMBLE METHOD USING USER DEFINED CUT-OFF LIMIT

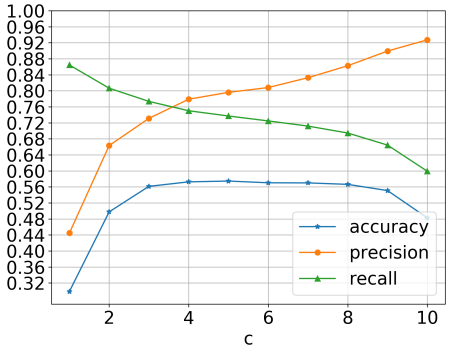
In the main text, we utilize WOC (Marbach et al., 2012) as consensus method for the ensemble variant of our approach. As it can be extracted from the presented results, this ensemble method achieves the best results in almost all evaluations and for all datasets. Yet, WOC cannot be parameterized in order to tend to a high precision or high recall outcome depending on the user's preference. For this reason, we also examine the use of a different consensus technique which lets the user define where should be the focus of the model. In this method we utilize Transformers equals to the number of known entity types. We have an user defined cut-off limit to indicate how many of them should at least predict a relation in order to include it in the final predictions set. Figure 6 presents the performance of this method for different cut-off limits in the three datasets. This method performs equally well with the WOC technique and it has the advantage that we can focus on high precision if c is close to the total number of available entity types or high recall if c is close to 0.



(a) Performance of Transformer ensemble variant for WebNLG dataset



(b) Performance of Transformer ensemble variant for NYT dataset



(c) Performance of Transformer ensemble variant for DocRED dataset

Figure 6: Performance of the Trasnformer ensemble variant in the three datasets. For each dataset, it is utilized as many Trasnformers as the number of different entity types and is examined the performance of the ensemble method for different cut-off limits c. The value of c indicates how many transformers at least should have predicted a specific relation in order to include in the final predictions set.

A.7 JENSEN-SHANNON DISTANCE

The Jensen-Shannon divergence metric between two probability vectors p and q is defined as,

$$\sqrt{\frac{D(p \parallel m) + D(q \parallel m)}{2}}$$

where m is the pointwise mean of p and q and D is the Kullback-Leibler divergence.

The Kullback-Leibler divergence for two probability vectors p and q of length n is defined as,

$$D(p \parallel q) = \sum_{i=1}^{n} p_i log_2(\frac{p_i}{q_i})$$

The Jensen–Shannon metric is bounded by 1, given that we use the base 2 logarithm.

A.8 ENTITIES EXTRACTION BASED ON ATTENTION ANALYSIS

In this section, we define the procedure that we follow in order to populate the actual KG based on the predicted relation types and the respective attention weights. For each instance, we generate an undirected graph that connects the tokens of the sentence based on their syntax dependencies. Then for each different predicted relation type, we define the final attention weights of a token based on the attention weights of itself and its neighbors in the syntax dependencies graph. Let a^r be the attention vector of a predefined model's attention head, which contains all the attention weights related to the relation type r. The final attention weight w of the token i for the relation r is defined as

$$w_i^r = 2 * a_i^r + \sum_{j \in neig_i} a_j^r$$

where $neig_i$ is the set containing all the neighbors of i in the syntax dependencies graph. Then for each noun chunk k (n_k) of the text we compute its total attention weight for the relation type r as:

$$n_k^r = \sum_{j \in nc_k} f(w_j^r)$$

where nc_k is the set of tokens which belong to the n_k and f is a function defined as

$$f(w_j^r) = \begin{cases} w_j^r & \text{if } j \text{ is stop-word} \\ 2 * w_j^r & \text{if } j \text{ is not stop-word} \end{cases}$$

Finally, we extract as entities which are connected via the relation type r the two noun chunks with the highest weight n^r. At this point and as this work is a proof of concept rather than an actual method, the selection of the attention head is based on whichever gives as the best outcome. Yet, in actual scenarios it is recommended the use of a training set, based on which the optimal head will be identified. For the creation of the syntax dependencies graph and the extraction of the noun chunks of the text we use spacy[1] and its `en_core_web_lg` pretrained model.

[1] https://spacy.io/