CEUR-WS.org/Vol-2941/paper4.pdf

Knowledge Graph Embeddings for News Article Tag Recommendation*†

Nora Engleitner[1], Werner Kreiner[2], Nicole Schwarz[2], Theodorich Kopetzky[2], and Lisa Ehrlinger[2,3]

[1] Newsadoo GmbH, Austria
nora@newsadoo.com
[2] Software Competence Center Hagenberg GmbH, Austria
firstname.lastname@scch.at
[3] Johannes Kepler University Linz, Austria
lisa.ehrlinger@jku.at

Abstract. Newsadoo is a media startup that provides news articles from different sources on a single platform. Users can create individual timelines, where they follow the latest development of a specific topic. To support the topic creation process, we developed an algorithm that automatically suggests related tags to a set of given reference tags. In this paper, we first introduce the Newsadoo tag recommendation system, which consists of three components: (1) item-based similarity, (2) knowledge graph similarity, and (3) actuality. We describe the knowledge graph component in more detail and analyze the suitability of different knowledge graphs and embedding techniques to enhance the quality of the overall Newsadoo tag recommendation. The paper concludes with a list of lessons learned and interesting future work.

Keywords: Knowledge Graph Embeddings · Tag Recommendation.

1 Introduction

Newsadoo[4] is a European media startup that provides articles from various regional, national, and international newspapers as well as magazines on a single platform. The aim is to keep users broadly and well informed while offering a certain degree of personalization to facilitate news consumption at the same time. In particular, users can select sources they trust and prefer to read, thereby influencing the news presented in their personalized timeline. Newsadoo further offers users the possibility to create individual timelines (so-called "topics") for their areas of interest, thereby staying up-to-date with the latest developments concerning a specific topic. These personal timelines can be generated either with

[*] The research reported in this paper has been funded by BMK, BMDW, and the Province of Upper Austria in the frame of the COMET Programme managed by FFG and the FFG General Programme project "TIDE", GA no. 880693.

[†] Copyright © 2021 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

[4] https://newsadoo.com (July 2021)

custom search terms or by selecting tags that are provided within Newsadoo. Tags represent keywords for an article and are extracted automatically by using a combination of named entity recognition (for detecting the keywords) and entity linking with Wikipedia and Wikidata (for obtaining uniform and unique tags).

To support the user in the topic creation process, we developed an algorithm that suggests related tags to a set of given reference tags. We obtain these tag recommendations by analyzing common tag occurrences in Newsadoo articles on the one hand, and by incorporating information from a public knowledge base on the other hand. Section 2 details on the tag recommendation system. In Section 3, we evaluate three existing knowledge graph (KG) embeddings as well as self-trained embeddings to increase the quality of automated tag recommendation.

2 The Newsadoo Tag Recommendation

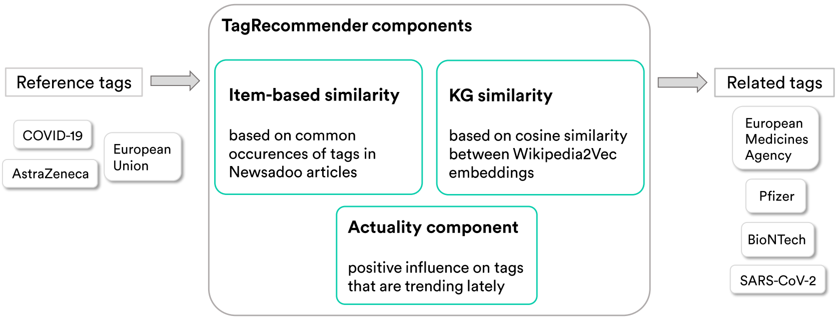
Reference tags
COVID-19
AstraZeneca
European Union
TagRecommender components
Item-based similarity
based on common occurences of tags in Newsadoo articles
KG similarity
based on cosine similarity between Wikipedia2Vec embeddings
Actuality component
positive influence on tags that are trending lately
Related tags
European Medicines Agency
Pfizer
BioNTech
SARS-CoV-2

Fig. 1. Newsadoo tag recommendation

Fig. 1 depicts a schematic representation of the Newsadoo tag recommendation system. As input, a set of reference tags (e.g., "COVID-19", "AstraZeneca", "European Union") is provided and as output, we obtain a ranked list of tags, which are related to the combination of the input tag set. The recommender itself is based on an ensemble algorithm, which uses the following three components:

- The *item-based similarity* (IBS) component evaluates which tags appear most often together with the reference tags in Newsadoo articles.
- The *knowledge graph similarity* (KGS) employs KG embeddings (containing the Newsadoo tags) to determine the most similar entities for the reference tags by computing the cosine similarity between the reference tags and tags in the KG. The development of the KGS is discussed in detail in Section 3.
- The *actuality component* increases the rating of tags, which appear more frequently in recent articles. Thus, the recommendation can be influenced by recent events, which is an important factor for a news platform.

The final tag recommendation result is obtained by merging the related tags provided by the IBS and KGS components, computing a combined similarity score for these tags and sorting the result accordingly.

3 A Comparison of KG Embedding Techniques

To select the most suitable KG embedding technique for the tag recommendation, we evaluated three existing KG embeddings: KGvec2go [5], Wembedder [4], and pre-trained embeddings from PyTorch–BigGraph [2]. Further, we trained our own embeddings using pyRdf2Vec [6] (based on Wikidata and DBpedia) and Wikipedia2Vec [8] (based on the German and English Wikipedia).

Pre-trained embeddings. We found that the existing embeddings from KGvec2go and Wembedder were not suitable for our application since the results were outdated or very unrelated to the input tags. The pre-trained embeddings from PyTorch–BigGraph performed generally well with the exception for location tags, where the results were often not relevant enough. Therefore, we decided to try another method and compute a self-trained KG embedding, which allows use-case-specific optimization, for our application.

Self-trained embeddings. For building our own embeddings, we experimented with Wikidata[5], DBpedia[6], and DBpedia Live[7]. All three KGs performed well for a small amount of items, but were not suited for practical application in Newsadoo. As there are tools to build dumps for Wikidata, and since DBpedia and DBpedia Live are language-specific, we focused on the language-independent Wikidata as Newsadoo offers news articles in different languages. With Wikidata the major challenge was to identify a suitable approach for creating embeddings for the vast amount of entities provided in this KG.

Available dumps could not be processed directly due to memory limitations. Accessing the online SPARQL endpoint[5] during training would have led to an evaluation time of several weeks. The effort to host an endpoint locally was considered disproportional high. Therefore, we built our own local Wikidata dump to train the embeddings locally. This subgraph was obtained by querying the SPARQL endpoint for each of the 400,000 items and restricting the result to triples containing a Wikidata item as object. We optimized our walking strategies and parameters according to the findings from [1] and [7]. The best results for a runtime of one day was achieved with the Weisfeiler-Lehman strategy, max. 100 walks per item, a walking depth of 4, and a vector size of 100.

These embeddings yielded generally good results for our application with a few exceptions: In some cases, the resulting items were too similar to each other, e.g., for a car manufacturer as reference tag we obtained a list of different car models from this manufacturer. This might be acceptable or even desirable for other applications, but in our case we require a certain diversity within the results. Additionally, we observed examples, where the result contained elements that would be considered irrelevant when using it for tag recommendation, e.g., “Austrian Sign Language” for the reference tag “Austria”.

[5] https://query.wikidata.org/sparql (July 2021)

[6] https://dbpedia.org/sparql (July 2021)

[7] https://live.dbpedia.org/sparql (July 2021)

Wikipedia2Vec. Due to the drawbacks mentioned above, we considered a third approach and created embeddings via Wikipedia2Vec. This model is strictly speaking not a KG embedding, but rather an embedding of regular vocabulary and Wikipedia entities into the same vector space via skip-gram based models [3]. More precisely, the Wikipedia2Vec model is trained by jointly optimizing three different models: one of these models utilizes the Wikipedia link graph and learns to predict neighboring entities in this graph. The second model is a conventional skip-gram model applied to the text on a Wikipedia page. The third model learns to predict neighboring words of a target entity and thereby places similar words and entities near to each other in the vector space.

Since there are currently English and German tags available in Newsadoo, we require embeddings for both languages and therefore combine the results for obtaining language-independent recommendations. Furthermore, we incorporate the frequency of an item, which is also computed during the embedding algorithm, into the similarity score to filter out less relevant entities.

Final decision. In real-world applications, it is generally challenging to determine the quality of the results, since typically, no annotated data is available. In addition, for our tag recommendation system, the quality of a result is highly subjective and dependent on the expectations of the user. Since user feedback was not available at the development stage, we decided to rely on the domain knowledge of experts for evaluating the quality of the results for this specific application. Therefore, we defined a representative set of reference tags and performed a qualitative evaluation of the top 10 recommended results for different embeddings. Table 1 shows a subset of this evaluation. Note that the set of feasible results is restricted to the set of available tags in Newsadoo.

Eventually, we decided to use Wikipedia2Vec as most suitable embedding for our application due to the following reasons: First, this approach provides consistently good results without any completely irrelevant tags as opposed to other models. Second, we found that Wikipedia2Vec yields a higher diversity than pure KG embeddings as discussed in the car manufacturer example above.

4 Conclusion and Research Outlook

In this paper, we introduced the Newsadoo tag recommendation system, which provides related tags to a set of given reference tags (with tags being special keywords extracted from a news article). One crucial component in this system are KG embeddings, which were investigated and evaluated with respect to tag recommendation in greater detail.

We found that Wikipedia2Vec delivered the best results (in terms of suitability and diversity) for our application based on a qualitative evaluation with domain experts. Preparing data for training was challenging due to (1) performance issues using online SPARQL endpoints within the training process, (2) memory limitations for available dumps, and (3) maintenance overhead with a locally hosted endpoint. For future work, we plan to extend the current solution with more research on the tuning of the subgraphs and an approach for

Table 1. Comparison of different KG embedding techniques for tag recommendation.

PBG	**Wikipedia2Vec (en+de)**	**pyRdf2Vec – Wikidata**
Austria		
Maissauer (noble family)	Germany	Vienna
State Gallery of Lower Austria	Switzerland	Switzerland
State Gallery of Lower Austria	Vienna	Municipality (Austria)
Klafferkessel	Tyrol (state)	Italy
Langschwarza	Styria	Hungary
EU-protected-area March-Thaya-Auen	France	Austrian Sign Language
Netflix		
Facebook Watch	Prime Video	Ask the StoryBots
Amazon Studios	Hulu	Amazon Web Services
Red Bull TV	Video on demand	Amazon (company)
YouTube Premium	Crunchyroll	Alliance for Open Media
Hulu	HBO	Big Mouth (TV series)
Set-top box	Streaming media	Hulu
BMW		
Volkswagen Group	Mercedes-Benz	BMW 6 Series
Daimler-Benz	Audi	BMW 1 Series
BMW Motorrad	Porsche	BMW Z
Chrysler LHS	BMW Motorrad	BMW GS
Cadillac	Volkswagen Group	BMW 320
Mercedes-Benz Cars	Volvo	BMW X1

evaluating the quality of the tag recommendation in greater detail, e.g., with an information-retrieval-style relevancy evaluation. We also plan to investigate the suitability of even more recent approaches, e.g., graph neural networks.

References

1. Iana, A., Paulheim, H.: More is not always better: The negative impact of a-box materialization on rdf2vec knowledge graph embeddings. arXiv:2009.00318 (2020)
2. Lerer, A., Wu, L., Shen, J., Lacroix, T., Wehrstedt, L., Bose, A., Peysakhovich, A.: Pytorch-biggraph: A large-scale graph embedding system. arXiv:1903.12287 (2019)
3. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv:1301.3781 (2013)
4. Nielsen, F.: Wembedder: Wikidata entity embedding web service. arXiv:1710.04099 (2017)
5. Portisch, J., Hladik, M., Paulheim, H.: KGvec2go – knowledge graph embeddings as a service. In: Proceedings of the 12th Language Resources and Evaluation Conference. pp. 5641–5647. European Language Resources Association, Marseille (2020)
6. Vandewiele, G., Steenwinckel, B., Agozzino, T., Weyns, M., Bonte, P., Ongenae, F., Turck, F.D.: pyRDF2Vec: Python implementation and extension of rdf2vec (2020), `https://github.com/IBCNServices/pyRDF2Vec` (July 2021)
7. Vandewiele, G., Steenwinckel, B., Bonte, P., Weyns, M., Paulheim, H., Ristoski, P., Turck, F.D., Ongenae, F.: Walk extraction strategies for node embeddings with rdf2vec in knowledge graphs. arXiv:2009.04404 (2020)
8. Yamada, I., Asai, A., Sakuma, J., Shindo, H., Takeda, H., Takefuji, Y., Matsumoto, Y.: Wikipedia2vec: An efficient toolkit for learning and visualizing the embeddings of words and entities from wikipedia. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. pp. 23–30. Association for Computational Linguistics (2020)