University *of Ljubljana*
Faculty *of Computer and Information Science*

Extending SloWNet using Machine Translation and Digital Dictionary Database

Miha Bombek, Luka Dragar, David Ocepek

Abstract

For our project, we extended the current SloWNet 3.1 (43,460 synsets) by translating missing synsets from the Princeton Wordnet (117,000 synsets) using the pre-trained Nvidia NeMo sequence-to-sequence transformer. We then linked the extended SloWNet synsets to the Slovenian Digital Dictionary Database (DDD) and added additional information from it, such as definitions as well as additional literals. The precision of our machine translated synsets was 0.42, this is a significant drop in precision from the original SloWNet's 0.91 and results in an average 0.50 translation precision. We extended SloWNet 3.1 with 40297 Slovene literals and 602 Slovene definitions.

Keywords

WordNet, machine translation, entity linking, word embeddings

Advisors: Assist. Prof. Dr. Slavko Žitnik, Assist. Aleš Žagar

Introduction

Wordnets are lexical knowledge graphs (KG) with many practical use cases, such as information retrieval, automatic text classification, machine translation (MT), etc. Despite their wide range of use cases, Wordnet construction is a difficult task often requiring manual annotation of tens of thousands of words by experts. Because of the aforementioned difficulties Wordnets are frequently built using MT models [2, 6]. For our project, we focused on extending the Slovene Wordnet (SloWNet) by translating and adding the synsets from the English Princeton Wordnet (PWN). We then added additional information, obtained from the DDD [1], such as definitions and new literals, to each synset. Under our proposed approach one can potentially almost triple the number of synsets in SloWNet while simultaneously using DDD for anomaly detection and quality control therefore utilizing already existing manually annotated Slovene resources to achieve a construction accuracy that is likely to be above that of heuristic methods that are used to construct Wordnets for low resource languages.

Methodology

SloWNet representation

We used the newest implementation of SloWNet (version 3.1) [2] as the basis for our project, keeping already present translations and only adding new translations generated by the NeMo model. SloWNet is primarily stored as an XML document which is hard to use and unintuitive, therefore we converted it to graph form. Each node represented a synset and contained the following information: PWN synset id, list of English literals that are part of the synset, and list of Slovene literals (if any) that are part of the synset. Edges represented the relationships between synsets such as whether two synsets were similar in meaning, had opposite meanings or whether one synset was a hyponym to another synset.

Machine translation

For translating synsets from the English WordNet to Slovene, we utilized the NeMo toolkit [3], developed by NVIDIA. For our purposes, we used a pre-trained model for Slovene-English translation [4], which was trained on approximately 33 million translation pairs. We implemented both single and list-based synset translation, translating each synset word individually or all of them together as a comma-separated list of literals. The latter proved to be more accurate than the prior in instances where it was applicable (where a synset included multiple English literals) likely because it provided context to the NeMo model. Additionally, we put each individual literal into single quotation marks as otherwise, the NeMo model exhibited weird and undesirable behavior such as adding arbitrary verbs to adjectives and translating literals in different declensions. Both problems are still present but their frequency was significantly diminished by the above approach.

Lemmatization of Slovene literals

English literals are occasionally translated into Slovene in non-standard declensions. While this does not seriously impact our precision it does lead to an increased multiplicity of certain literals and increases the difficulty of linking the DDD database with our SloWNet, therefore we used the language processing package CLASSLA [8] to standardize (lemmatize) our translations.

Translation confidence

Each word translation was assigned a prior translation confidence equal to its expected precision, for manually translated words the accuracy was 0.91 while for the machine-translated words, the accuracy was 0.42. We also translated each translated word back into English and increased the confidence by 0.10 if it translated back into the same synset.

Validation with word embeddings

Once the synsets were translated, we employed additional validation with the use of word embeddings. The idea was to test the semantic similarity of literals within the synsets and determine if they are really synonyms. The literals were tested pairwise for the similarity using the standard cosine similarity metric. The word embeddings used are available through the Clarin repository [5]. As we need two or more words to measure the similarity, synseys consisting of only one word could not be validated in this way. Each validated synset was assigned a confidence equal to the average similarity between each pair of literals contained in it. Word embeddings could potentially also be used for attaching new words, as we can find the synset containing literals that are most similar to the target word.

DDD Linking

To extend SloWNet with data from DDD, we first linked synsets from SloWNet to corresponding senses from DDD. Firstly, we exported relevant data from DDD into a separate knowledge base. For each sense, we exported the category (part-of-speech tag) and the lemmatized form of its lexical unit. We also included the forms of lexical units of related senses. These related senses include synonyms, antonyms, hypernyms and hyponyms. The new knowledge base contains data for 199079 senses from DDD.

We then iterated over all of the synsets from the translated SloWNet and for each synset, we iterate over all of its literals. For every literal, we choose senses from DDD, whose lexical units have the same form and the same category as the literal, as possible candidates. For each candidate, we then check its synonyms, antonyms and hypernyms. For every candidate, we also check whether it is a so called *dummy sense* or not. Dummy senses are senses from DDD, that contain data, that often belongs to several different senses, but it hasn't been manually distributed yet.

For single-literal synsets, we create a link with all candidates, since the literal has no synonyms to match by. These links are the weakest. For synsets with more than one literal, we create a link with all candidates, that share at least one synonym, antonym or hypernym. These links are stronger and much more likely to be correct. The strongest links are those, where the linked DDD sense is not a dummy sense.

To further extend the translated SloWNet with data from DDD, we only take the strongest links. For every linked synset, we take the synonyms from the linked DDD senses and include them in the synset as new literals. We also include definitions from DDD and add them to the synset.

Evaluation

We evaluated our extended SloWNet manually, as follows: firstly we randomly sampled 50 synsets, then for each Slovene literal we decided whether the literal matched the concept of the synset it was assigned too. We then calculated the precision as the ratio of literals that we consider to accurately represent that concept (synset) relative to the number of all Slovene literals assigned to that concept. Using the calculated precisions for the set of sampled concepts we then calculated the bootstrapped mean and standard deviation of our expected precision (Eq. 1).

$$\begin{aligned}
&\forall s \in S: \operatorname{precision}(s) = \frac{\textit{words belonging to synset } s}{\textit{words assigned to synset } s}, \\
&\forall \hat{S} \in \operatorname{bootstrap(S)}: \hat{\mu}_{\text{precision}} = \frac{1}{|\hat{S}|} \sum_{s \in \hat{S}} \operatorname{precision}(s), \\
&\mu_{\text{precision}} = E[\hat{\mu}_{\text{precision}}], \quad \sigma_{\text{precision}} = D[\hat{\mu}_{\text{precision}}]. \qquad (1)
\end{aligned}$$

Results

Comparison to SloWNet 3.1

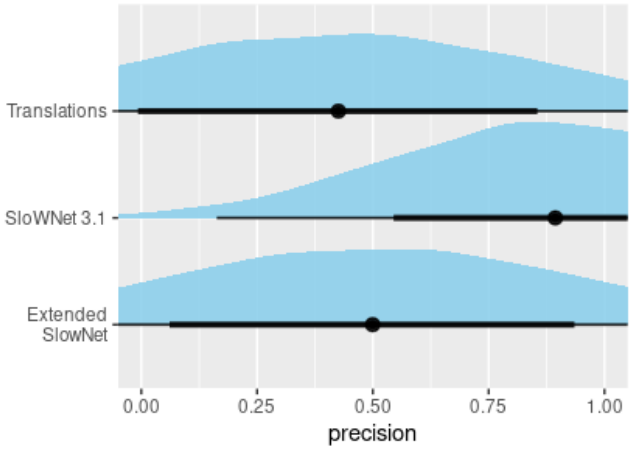

Figure 1. Precision of the SloWNet 3.1, our translations, and the extended SloWNet.

The overall mean precision of our extended SloWNet was $\mu = 0.50$ with a standard deviation of $\sigma = 0.46$ (Fig. 1), however, it is important to note the precision of our machine-translated synsets was only $\mu = 0.42$, $\sigma = 0.45$. This is significantly lower than the manual translation precision of $\mu = 0.91$, $\sigma = 0.37$. In order to encode the difference between machine and manual translations we assigned each synset a confidence score matching its precision the type of translations precision

which should enable manual translators to prioritize certain more precisely translated synsets over others.

DDD linking analysis

With our linking method, we obtained 135457 links between DDD senses and SloWnet synsets. These contain 42499 distinct DDD senses, linked to 36524 distinct synsets. It was expected, that on average, several senses would be linked to the same synset, because of the data structure in DDD. Each literal from the synsets should have its own sense in DDD, so synsets can contain several senses. Of all the links, we have 18745 strongest links, which we use to extend SloWNet with DDD data.

However in theory, the number of links should be the same as the number of distinct DDD senses linked, since each sense should be linked to at most one synset. This is not the case because of several factors. The most prevalent one is, that not all senses of lexical units in DDD are manually checked and there exist dummy senses. Because they contain data, including synonyms, that should be distributed among different senses, they can be linked to several different synsets.

By combining the obtained translations with additional DDD data, we were able to extend SloWNet 3.1 with 40297 new Slovene literals and 602 new Slovene definitions.

Discussion

Anomaly detection using DDD

By linking entities from DDD and SloWNet, we could better evaluate the quality of translations and maybe even improve translation accuracy. For example, if we get a link between a DDD sense and a SloWNet synset by matching synonyms, there is a high probability, that the matching literals were translated correctly so that they still remained synonyms in Slovene, in order to be matched with synonyms from DDD. With this approach, we could introduce some probability that a translation is correct for every literal.

We could also remove literal translations in linked synsets, that didn't match any of the synonyms from DDD. This would improve the overall accuracy of the translations, but we could lose some correct translations, due to them not being listed as synonyms in DDD.

Automatic evaluation using word embeddings

Word embeddings are a popular tool for comparing the similarity of words and can be used to validate the words within synsets, as we have used it here. Recently, a new method for converting a lexical semantic graph to lexical semantic space was proposed [7] which could be used in conjunction with a standard similarity prediction task to obtain a more thorough evaluation of newly constructed synsets. There is an analogy dataset available for Slovene [9] that could be used to evaluate our constructed embeddings. We note however that this method is only suitable for evaluating the synset validity but cannot evaluate the relations between them.

Additional DDD data

With the obtained links between DDD senses and SloWNet synsets, we can add several types of data from DDD to the synsets. We only added new literals and definitions. DDD also contains a lot of usage examples for the senses, from the Gigafida 2.0 corpus. We weren't able to add those because of strict licensing on Gigafida 2.0. In the future, additional data from DDD could be included in SloWNet, based on the obtained links.

References

[1] CJVT. *Digital Dictionary Database*. Accessed on 16. 3. 2023. 2023. URL: https://wiki.cjvt.si/books/digital-dictionary-database.

[2] Darja Fišer and Tomaž Erjavec. *sloWNet: construction and corpus annotation*.

[3] Oleksii Kuchaiev et al. "NeMo: a toolkit for building AI applications using Neural Modules". In: *CoRR* abs/1909.09577 (2019). arXiv: 1909.09577. URL: http://arxiv.org/abs/1909.09577.

[4] Iztok Lebar Bajec et al. *Neural Machine Translation model for Slovene-English language pair RSDO-DS4-NMT 1.2.6*. Slovenian language resource repository CLARIN.SI. 2022. URL: http://hdl.handle.net/11356/1736.

[5] Nikola Ljubešić and Tomaž Erjavec. *Word embeddings CLARIN.SI-embed.sl 1.0*. Slovenian language resource repository CLARIN.SI. 2018. URL: http://hdl.handle.net/11356/1204.

[6] Roberto Navigli and Simone Paolo Ponzetto. "BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network". In: *Artificial intelligence* 193 (2012), pp. 217–250.

[7] Chakaveh Saedi et al. "Wordnet embeddings". In: *Proceedings of the Third Workshop on Representation Learning for NLP* (2018). DOI: 10.18653/v1/W18-3016.

[8] Luka Terčon, Jaka Čibej, and Nikola Ljubešić. *The CLASSLA-Stanza model for lemmatisation of standard Slovenian 2.0*. Slovenian language resource repository CLARIN.SI. 2023. URL: http://hdl.handle.net/11356/1768.

[9] Matej Ulčar et al. "Multilingual Culture-Independent Word Analogy Datasets". In: *Proceedings of the Twelfth Language Resources and Evaluation Conference* (2020).