StarNet: A WordNet Editor Interface

Ezgi Sanıyar, Oğuzhan Kuyrukçu
Starlang Yazılım Danışmanlık
ezgi.saniyar@gmail.com, kuyrukcuoguz@gmail.com

Olcay Taner Yıldız
Özyeğin University
olcay.yildiz@ozyegin.edu.tr

Abstract

In this paper, we introduce StarNet WordNet Editor, an open-source annotation tool designed for natural language processing. It's mainly used for creating and maintaining machine-readable dictionaries like WordNet (Miller, 1995) or domain-specific dictionaries. WordNet editor provides a user friendly interface and since it is open-source, it is easy to use and develop. Besides English and Turkish WordNet (KeNet) (Bakay et al., 2020), it is also applicable to several languages and their domain specific dictionaries.

1 Introduction

Wordnets are natural language processing resources used in tasks like information retrieval or categorization. As a broad definition, a WordNet is a machine-readable dictionary with the lexicographic information of words including synsets and separate senses of those synsets. Mainly, synsets are the single units that the semantic relations or mappings are built on. Senses, on the other hand, are the definitions given for each synset. Based on the idea that words can be explained by their relations to other words, WordNets offer basic semantic relations such as hypernymy, meronymy, or antonymy between synsets.

After the Princeton WordNet (PWN) (Miller, 1995) several WordNets in different languages have been created. Finnish WordNet FinnWord- Net (Lindén and Carlson, 2010), Polish Word-Net (Derwojedowa et al., 2008) , and French WordNet WOLF (Sagot and Fiser, 2008) are some of the pioneering WordNets in the world. The multilingual WordNet EuroWordNet (EWN) (Vossen, 1997) is another significant WordNet comprising seven European languages, namely English, Dutch, Italian, Spanish, German, French, Czech, and Estonian. Turkish, on the other hand, has mainly two WordNets; BalkaNet (TR-WordNet) (Bilgin et al., 2004) and KeNet (Bakay et al., 2020). TR-Wordnet of BalkaNet is the first WordNet of Turkish and has 14,626 synsets, while KeNet is currently the largest Turkish WordNet with 76,757 synsets.

Wordnets can be presented or edited by software (editors) designed for this purpose. These editors are used to edit WordNets and they have a crucial role in correcting or updating items, matching synsets with their synonyms and composing semantic relations like hypernyms.

StarNet WordNet Editor is one such software. It is designed to perform multipurpose functions in order to build, edit and group synsets and senses in WordNets. It has been primarily used for Turkish and English. However, it is suitable to be used for any target language, regardless of the morphological complexity of the language across the analytic-synthetic spectrum. In this paper, we present our multi-functional WordNet editor StarNet and discuss each of its functions and process applied to it. We present a literature review on editors in section 2, describe and discuss the functions of each component of our editor in sections 3 and 4, and present a conclusion in section 5.

2 Literature Review

Originally intended to be manually consulted, the purpose of Wordnets turned more towards automatic processing, and a need for interfaces to connect this resource onto different applications was born (Tufis et al., 2004). Visdic, developed by the team of Czech WordNet (Horák and Smrž, 2004) and Polaris (Louw, 1997) and Periscope (Cuypers and Adriaens, 1997), employed by EuroWordNet are examples of softwares designed for this purpose. Visdic is used for presenting and editing dictionaries stored in XML format and it's configurable with regards to program behaviour and dictionary design. Polaris is used to create and edit WordNets, while Periscope is used to view said WordNets. Both are in addition used to export WordNets. However, when it comes to building

WordNets from scratch, these softwares are not very convenient options. Polaris is a licensed and rather expensive software that is no longer being developed and Visdic is not optimized for building but rather presenting & editing WordNets. Here we present a new, easy to use and open source alternative that can be used effectively to build new WordNets as well and view and edit existing ones.

In creating and mapping WordNets, two main approaches are being used; the expand approach and the merge approach. The expand approach takes PWN as the base and translates it to the target language (Vossen, 1996). Once the relations are transferred from English, they are checked manually. French (Sagot and Fiser, 2008) and Finnish (Lindén and Carlson, 2010) WordNets are examples of the expand approach. On the other hand, in the merge approach, PWN/English WordNet is not taken as the base. WordNets are created independently with intra-lingual relations and these are then linked to English. Our approach is based on the merge approach like Polish WordNet (Derwojedowa et al., 2008), Russian WordNet (Balkova et al., 2004), Norwegian WordNet NorNet (Fjeld and Nygaard, 2009) and Danish WordNet DanNet (Pedersen et al., 2009). The expand approach is assumed to be a practical way for building a new WordNet in target languages, but it may be biased towards the imitated WordNet. Merge approach, on the other hand, results in more concrete and accurate structures for languages that differ from English in their semantic patterns and potentially allows us to maintain language-specific properties (Bakay et al., 2020), (Vossen et al., 1998).

We used five different editors for different components of a WordNet. This allows the user to modify these components independently of each other. Our program works with XML format. It works as a desktop application and employs Java for back end structure. It can thus be used with all major operating systems. In the following, we will explain how our program works component by component.

3 Editors

3.1 Literal Matcher

The construction of the synsets presented and edited in our interface is derived from the latest Contemporary Dictionary of Turkish (CDT) (Ehsani et al., 2018) published by the Turkish Language Institute (TLI). In the dictionary, it is stated that the synonym literals are mainly used in the definitions of senses, which are given in one line separated with commas. For example, the definition of word *kırmızı* (red), is 'Kırmızı renkte olan, kızıl, al' (Something in red color); and possible synonym literals of word *kırmızı* are *kızıl* and *al*. After extracting possible synonym literals from the definitions, they are annotated by human annotators. In this part of the process, the Literal Matcher is a great help in viewing the literals that are possible synonyms in a synset.

The Literal Matcher is a tool enabling synonym literal matching in the target languages. This interface offers many facilities such as presenting every sense definition of a unique literal, convenient editing and a quick tag-save mode, which saves processes as soon as literals are matched, without further operation (Figure 1). Synonym candidates will appear in two groups in this component. The interface enables us to annotate and match approximately 250 synset literals in an hour. While the tool is easy to use and practical in many ways, checking multiple meanings and synonyms in every step can decrease the speed of the matching process.

The Literal Matcher is a practical option for matching intralingual synonym literals. However, transitivity may cause problems as a result of multimatching. Even if the first literal and the second literal sense definitions are completely synonymous, when these literal matches are prolonged, the first literal definition and the fourth/fifth literal definitions may not be exactly synonymous. As a solution to this problem, StarNet presents the editor Synset Matcher. Such overgrown synsets with weak or absent synonym relations between its literals can be viewed and edited in the Synset Matcher by using split/merge processes.

3.2 Synset Matcher

As mentioned above, creating synsets with synonym literals can be challenging especially when the mapping is overgrown, the transitivity decreases. This process poses a problem in creating meaningful and accurate synsets. Here, the Synset Matcher plays a crucial role as it enables us to view all the literals in synsets and merge/split the synsets when necessary.

The Synset Matcher receives data from the Literal Matcher and acts as a supportive editor. It provides editing options for synonym literals in languages and provides an easy and practical interface

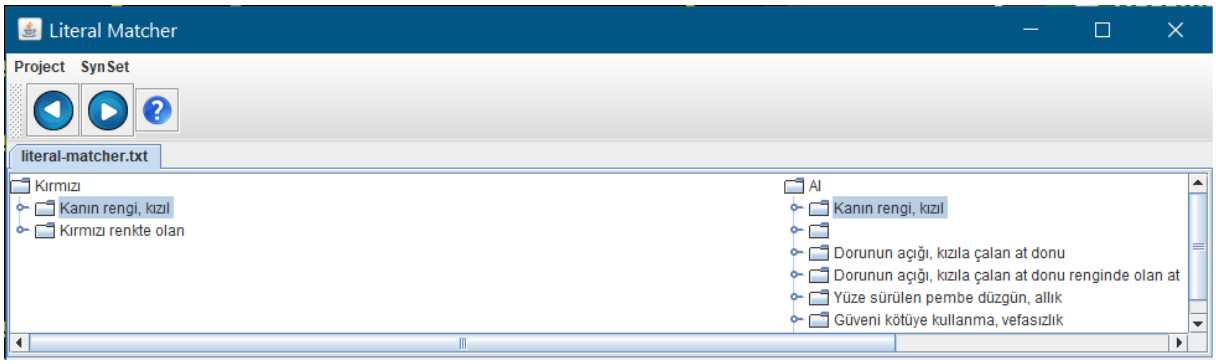

Figure 1: Interface of Literal Matcher with the synonyms of *red* in Turkish

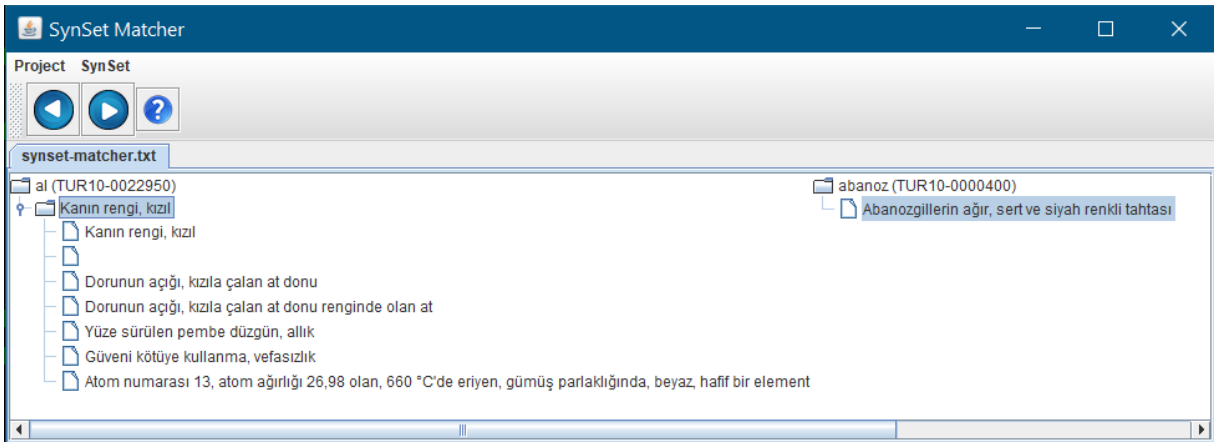

Figure 2: Interface of Synset Matcher; the first and final match example of red

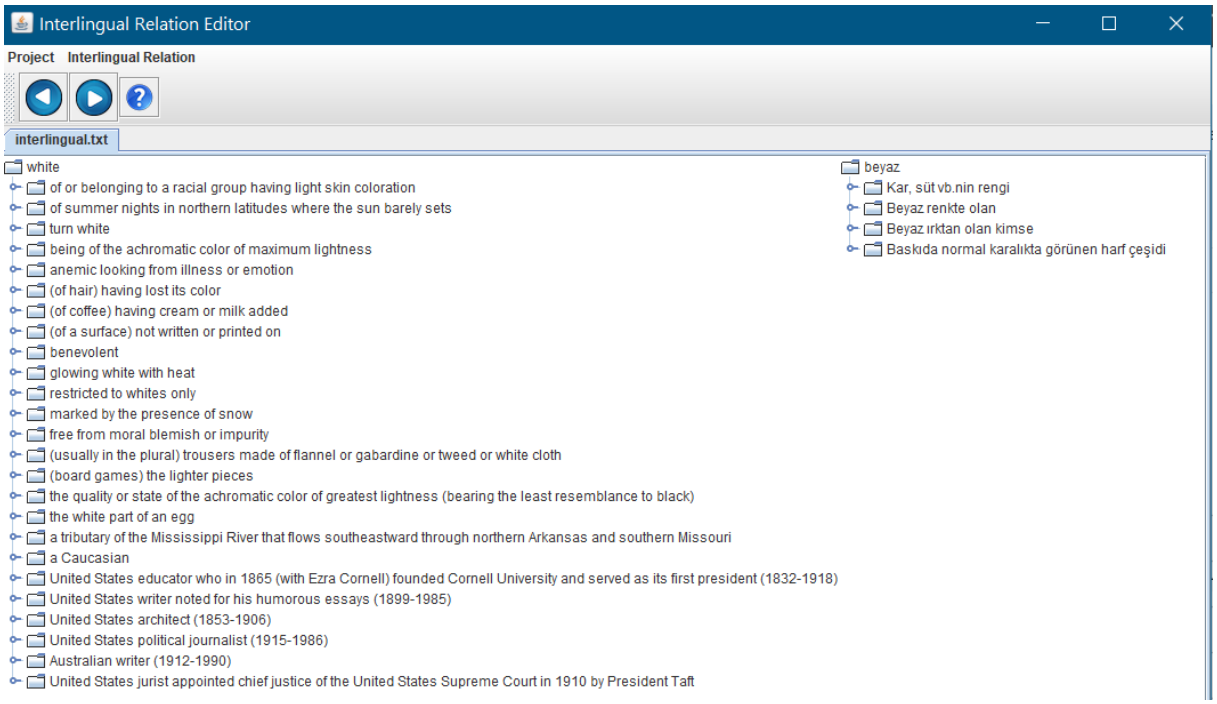

Figure 3: Interface of Interlingual Matcher with the English-Turkish synsets of white

to check the synsets built in the Literal Matcher. It allows us to identify the different synsets that should be grouped together because of their meanings and enables us to merge them. Similarly, any

synsets whose literals should be separated because of their unrelated definitions that are grouped together as a result of transitivity problems or any other mistakes during the previous processes can be split via the Synset Matcher. The Synset Matcher makes it possible to see the whole picture of a synset by showing us the final matching maps of all of its literals and to prune the synset if need be. As a result of this mapping and editing process in the Synset Matcher, we obtain the final version of synsets (Figure 2).

3.3 Interlingual Matcher

Interlingual relations and matching have great importance in the development of WordNets since creating these relations and linking the WordNets of different relations provide us with an important resource in many areas like machine translation. Therefore, an editor that works interlingually is a crucial tool in creating internationally applicable and useful resources and connecting the created WordNets to each other.

StarNet WordNet editor has an interface that enables inter-lingual matching. In creating KeNet, a merge approach is used and synsets in KeNet and PWN are matched as a result of this merging process. Both the synset matches and possible multilingual relations are checked by human annotators. The synset groups created in this process are transferred to the Interlingual Matcher to view and edit the matches.

The Interlingual Matcher is used by English PWN and Turkish KeNet data and matched synsets one-to-one between the languages by human annotators. As a result of this process, the existing matches can be checked and confirmed, and new matches can be created when needed. This process is potentially applicable to all languages via the Interlingual Matcher.

The Interlingual Matcher interface is quite similar to the Literal Matcher's interface and is easy to understand. The tag-save mode is active for the Interlingual Matcher as well. Unlike the Literal Matcher, however, only one-to-one matching is offered in the Interlingual Matcher: For each English word, suggested synonyms from the other language can be chosen and tagged (Figure 3).

3.4 WordNet Hypernym Editor

The WordNet Hypernym Editor provides an interface to build semantic hierarchies between synsets. With this component, we can annotate synsets in separate categories through semantic relations. This interface has enabled us to create our hypernym relations, and has been providing great convenience in other ongoing projects (Figure 4) like Turkish Estate WordNet and Turkish Tourism WordNet. Figure 4 shows us the interface of the hypernym editor and synsets derived from domain-specific Turkish WordNets.

The WordNet Hypernym Editor toolbar provides us with the opportunity to quickly and practically execute all the operations we might need to perform in the dictionary. It has options such as "quick save", "edit", "insert child", "remove child" (see below for *child*), "merge" or "change font size" (which may prove important for the well-being of the annotators' eyes). In addition, it includes the options "add to WordNet from dictionary" and "add to dictionary from WordNet" that enables editing via WordNet and matching the dictionary with the WordNet of the language. Senses are at the forefront in this component and fast access to them is of great importance. For this reason, all synsets can be reached easily with all their senses. When we type literals in the search bar, we can see all the senses of that literal and organize hypernym relations according to the senses (Figure 5). The WordNet Hypernym Editor provides two operations, merge and split: During or after the editing phase, synsets that should be grouped with the same unique sense can be merged, or incorrectly combined synsets (such as those originating from meaning-related drifts or POS-related drifts (Bakay et al., 2019)) can be split.

Taking the PWN editing style (Miller et al., 1990) as an example, the WordNet Hypernym Editor allows us to organize words in four categories: noun, verb, adjective and adverb. This allows obtaining a synset tree similar to the English WordNet (Miller, 1995).

It should be noted that there would be too many items in a natural language dictionary to organize into a sensible semantic hierarchy on-the-go for the annotators. At least the upper levels of the intended hierarchy would need to be specified outside the program and serve as a guide for the annotators. Of course, the more comprehensive this guide hierarchy, the better; but majority of lexical items in a language would still need to be put in its proper place in the hierarchy by the annotators. Principles for placing individual senses into the hierarchy should be specified. However, since annotators will

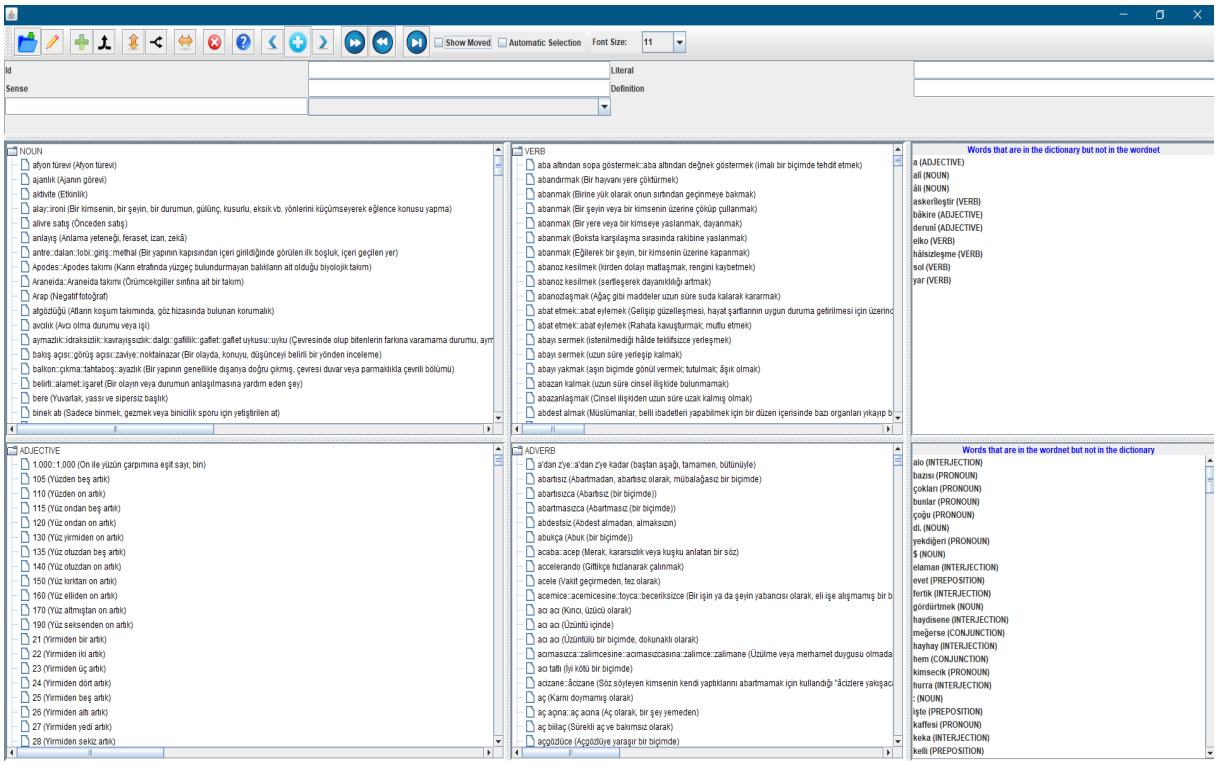

Figure 4: Interface of Hypernym Editor

have different understanding of some senses, there will inescapably be some subjectivity in the hierarchy that results, even if the annotators follow the same principles.

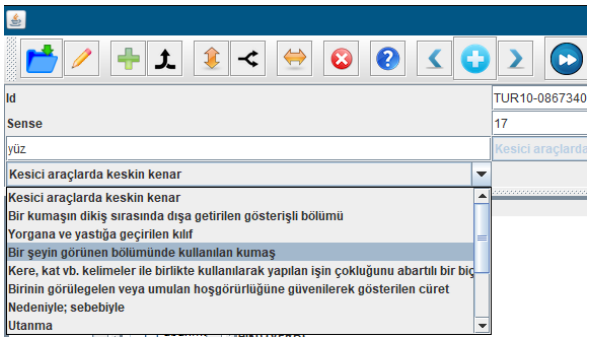

Figure 5: Synset Sense search of yüz which has different senses such as face, side, or part of something

Within the semantic hierarchy, we refer to a synset that is placed under another as a "child", and a synset with another synset placed below it is referred to as a "parent". In a completed hierarchy, every noun synset, except for the one designated at the top of the hierarchy (for example, in our WordNet, KeNet, this was *varlık* (being/entity), will have a parent. This is not necessitated by the editor, so if desired, items can be left out of the hierarchy or the dictionary could contain multiple independent hierarchies. Except for the end nodes, every synset will also have a child or children. Importantly, some synsets could have multiple parents. This is a rarer occurrence but natural languages might exhibit such semantic relations. *Su* (water) for example, is a child of both *sıvı* (liquid) and *ikili bileşik* (binary compound) in our WordNet. Of course, a synset could be assigned multiple parents by mistake too. When a synset has multiple parents, our editor will show it in red colors to distinguish it, so that it is easy to find them and correct their hypernym relations if necessary. Overall, since it has a practical interface, the WordNet Hypernym Editor allows an annotator to match approximately 70-80 synsets in one hour.

3.5 Dictionary Editor

The Dictionary Editor is distinct from the previous components in that it is an interface designed to create domain-specific dictionaries, whereas the former components are for building and maintaining natural language dictionaries. With the Dictionary Editor, synsets inside a WordNet can be added or removed and sense inputs of synsets can be edited in order to obtain a domain-specific dictionary. Whichever sense of a synset in the WordNet is used in that domain can be selected and

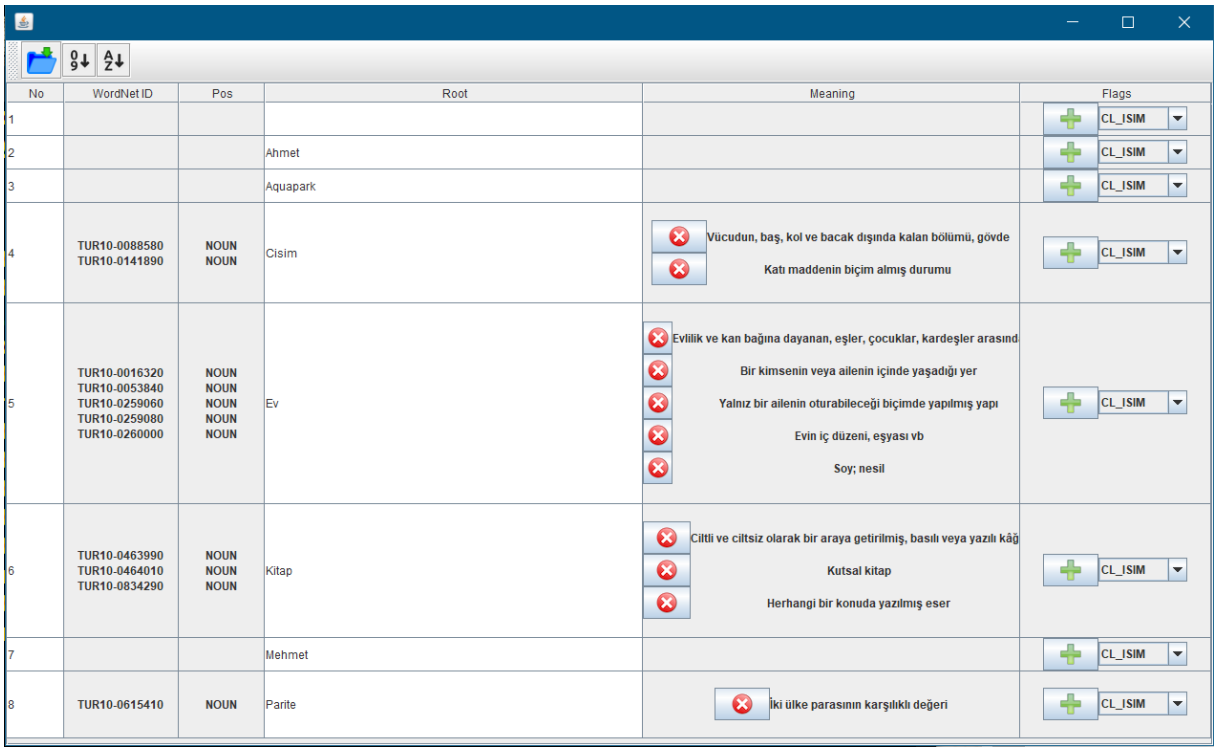

Figure 6: Interface of Dictionary Editor

transferred to the new dictionary or synsets can be transferred automatically from an existing WordNet to the domain-specific dictionary. Finally, if the sought sense is lacking, it can simply be added to the dictionary with this editor. This interface also makes sure that the dictionary and the WordNet are in accord: When an entry is added to the dictionary, it will be added to the WordNet too, and vice versa. The editor can also sort synsets numerically or alphabetically. The Dictionary Editor can be a practical tool for improving applications such as chat-bots or search engines. With the Dictionary editor, we have created several domain specific dictionaries including Turkish Estate WordNet and Turkish Tourism WordNet mentioned above. See (Figure 6) for the Dictionary Editor interface.

4 Discussion

StarNet WordNet Editor stands as a robust and open source alternative for people looking to develop a new WordNet. It can be used to view and build a domain-specific WordNet as well as a WordNet for a new target language. Being especially suitable for the merge approach, our editor will allow users to create new WordNets that preserve the language-specific features, which is especially important for agglutinative languages such as Turkish. Our editor also allows direct matching between WordNet and the morphological analyzer. Works on agglutinative languages such as Turkish or Hungarian, which may require exhaustive accuracy in morphological analysis for some expressions to be processed correctly in the WordNet, can particularly benefit from this feature. WordNet editor can be used on any operating system that supports Java, including Windows, Linux and Mac OS. It is in this regard unique among open source tools developed as a WordNet interface. In addition to being available and having advantages for various platforms and languages, WordNet Editor will present a user friendly interface for editing and maintaining a WordNet.

5 Conclusion

In this paper, we introduced a multipurpose editor. The editor we present has features that can be useful in establishing accurate synonym/hypernym relations and building domain-specific dictionaries. For future work, we intend to use it in other target language WordNets and incorporate Turkish FrameNet (Baker et al., 1998) (Marsan et al., 2021) into this editor and make it able to create and edit frame relations of languages.

References

Ozge Bakay, Ozlem Ergelen, Elif Sarmis, Selin Yıldırım, Atilla Kocabalcioglu, Bilge Nas Arican, Merve Ozcelik, Ezgi Saniyar, Oguzhan Kuyrukcu, Begüm Avar, and Olcay Taner Yıldız. 2020. Turkish WordNet KeNet. In *Proceedings of GWC 2021*.

Ozge Bakay, Ozlem Ergelen, and Olcay Taner Yildiz. 2019. Problems caused by semantic drift in wordnet synset construction. In *2019 4th International Conference on Computer Science and Engineering (UBMK)*, pages 1–5.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The berkeley framenet project. ACL '98/COLING '98, page 86–90, USA. Association for Computational Linguistics.

Valentina Balkova, Andrey Sukhonogov, and Sergey Yablonsky. 2004. Russian wordnet. In *Proceedings of the Second Global Wordnet Conference*, pages 31–38.

O. Bilgin, O. Cetinoglu, and K. Oflazer. 2004. Building a wordnet for Turkish. *Romanian Journal of Information Science*, 7:163–172.

I Cuypers and G Adriaens. 1997. Periscope: the ewn viewer. *EuroWordNet Project LE4003, Deliverable D008d012. University of Amsterdam*.

M. Derwojedowa, M. Piasecki, S. Szpakowicz, M. Zawisławska, and B. Broda. 2008. Words, Concepts and Relations in the Construction of Polish WordNet. In *Proceedings of GWC 2008*, pages 162–177.

R. Ehsani, E. Solak, and O.T. Yildiz. 2018. Constructing a WordNet for Turkish Using Manual and Automatic Annotation. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 17(3).

Ruth Vatvedt Fjeld and Lars Nygaard. 2009. Nornet—a monolingual wordnet of modern norwegian.

Ales Horák and Pavel Smrž. 2004. Visdic–wordnet browsing and editing tool. In *Proceedings of the Second International WordNet Conference–GWC*, pages 136–141.

Krister Lindén and Lauri Carlson. 2010. Finnwordnet–finnish wordnet by translation. *LexicoNordica–Nordic Journal of Lexicography*, 17:119–140.

M Louw. 1997. The polaris user manual. Technical report, Internal Report, Lermout & Hauspie.

B. Marsan, N. Kara, M. Ozcelik, B. N. Arican, N. Cesur, A. Kuzgun, E. Saniyar, O. Kuyrukcu, and O. T. Yıldız. 2021. Building the Turkish FrameNet. In *Proceedings of GWC 2021*.

George A. Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38:39–41.

George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. 1990. Introduction to wordnet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244.

Bolette Sandford Pedersen, Sanni Nimb, Jørg Asmussen, Nicolai Hartvig Sørensen, Lars Trap-Jensen, and Henrik Lorentzen. 2009. Dannet: the challenge of compiling a wordnet for danish by reusing a monolingual dictionary. *Language resources and evaluation*, 43(3):269–299.

Benoît Sagot and Darja Fiser. 2008. Building a free french wordnet from multilingual resources.

Dan Tufis, D Cristeau, and Sofia Stamou. 2004. Balkanet: Aims, methods, results and perspectives – a general overview. *Romanian Journal of Information Science and Technology Special Issue*, 7:9–43.

Piek Vossen, Laura Bloksma, Paul Boersma, M. Verdejo, Julio Gonzalo, Horacio Madrid, German Rodriquez, Politecnica Rigau, Barcelona Catalunya, Nicoletta Calzolari, Carol Peters, Eugenio Picchi, Simonetta Montemagni, and Wim Peters. 1998. Eurowordnet tools and resources report.

PJTM Vossen. 1996. Right or wrong: combing lexical resources in the eurowordnet project. In *M. Gellerstam, J. Jarborg, S. Malmgren, K. Noren, L. Rogstrom, CR Papmehl, Proceedings of Euralex-96, Goetheborg, 1996*, pages 715–728. Vrije Universiteit.

P.J.T.M. Vossen. 1997. Eurowordnet: a multilingual database for information retrieval. In *Proceedings of the DELOS workshop on Cross-language Information Retrieval, March 5-7, 1997 Zurich*. Vrije Universiteit. Proceedings of the DELOS workshop on Cross-language Information Retrieval, March 5-7, 1997, Zurich.