Semantic Network Integrity Maintenance via Heuristic Semi-Automatic Tests

Tomáš Čapek

Faculty of Informatics, Masaryk University, Brno, Czech Republic
xcapek1@aurora.fi.muni.cz

Abstract. In this article we discuss issues connected with maintaining content integrity of general-purpose semantic network that is in development. Construction of a semantic network from scratch is a long process that usually requires both linguistic work done by hand and semi-automatic methods to add or translate the data which must be subsequently reviewed. In this process many systemic and/or language-specific errors may appear in the data over time. We will introduce a method to cope with this issue systematically.

Key words: semantic network; WordNet; semantic network integrity

1 Introduction

A general-purpose semantic network is a language resource for the given language, alternative to traditional dictionaries. It consists of semantic units which are connected by semantic relations, thus creating a graph-like structure or a network. The biggest semantic network to date is WordNet (PWN) [1], that has been in development since 1985 at Princeton University. It contains more than 90.000 semantic units called synsets or synonymical sets. Many WordNet-like semantic networks exist today for other languages, developed in projects such as EuroWordNet [2] or BalkaNet [3].

To create a semantic network requires a team of linguists, software support and months of work among other things. In order to save time or resources one or both methods described below get usually employed:

- Semi-automatic translation of semantic units from other, larger networks. This also refers to so-called *expand model* in EuroWordNet terminology [4]. Basically it means that we adopt the original structure of semantic units and semantic relations among them, translate each lexeme automatically via some electronic dictionary or translator system available for our language and review the data afterwards by hand. Additional language-specific and other data are subsequently added to the network, thus *expanding* it. This is generally the fastest and the most popular method when creating a new wordnet-like semantic network and PWN is the most common semantic network used as a template. This method is the also most prone to adopting and creating new errors when used as the only method.

Petr Sojka, Aleš Horák (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2009, pp. 63–67, 2009.

- Manual linguistic work – also called *merge model*[1] in EuroWordNet terminology. The main focus here is to create a semantic network with independent structure or predetermined application in mind. An existing language resource can be used as the source lexicon, data in which need to be rearranged and interconnected via semantic relations to form a semantic network. This method is similar to traditional construction of dictionaries which is known to be time-consuming and expensive. It also requires a lot of linguistic introspection on part of the developers and can be outsourced so that many different people take turn in the process of adding and editing the data. As an implication semantic networks built according to this method are also prone to contain many types of inconsistencies and errors.

2 In-development Integrity Checks

There are several ways we can take to prevent errors from appearing in our network while it is still in development. However they represent additional expenses on time, resources and manpower. As evidenced in relevant Global WordNet Conference (GWC) proceedings articles [5], these additional methods have not been used more often than they have.

2.1 Corpus Evidence

When adding, checking or translating lexemes and semantic units it is important to have an appropriate corpus available as the definitive source of real-life usage of words. No two linguists have exactly the same knowledge and perspective of a language and that changes even for a single linguist over time. In this regard, corpora help to streamline and unify otherwise divergent approaches to handle linguistic data, especially those of non-frequent nature. The bigger the corpus is the better but it is also important for it to contain only relevant documents with respect to the contents of the semantic network. Unsorted pile of random documents can provide false or inaccurate evidence for the linguists thus spoiling the benefits corpora can bring to the process of development of a semantic network.

2.2 Guideline Manual

A set of instructions how to handle new or existing semantic units and relations sets the standard for people who participate in the semantic network development and who may come and go as the process goes on. It should provide basic information on issues such as: what are the criteria for a word to be lexicalized or non-lexicalized in the network; in what way to compose or

[1] EuroWordNet was a project primarily focused to create a multi-lingual semantic network based on PWN. The *merge* part of the process refers to the final stage of development when the semantic units in the newly created network are connected to their corresponding counterparts in another network, thus *merging* it into one bilingual structure.

assume definitions for semantic units; how to use notes for further work; what semantic relations are important for particular part of speech, etc. The nature of the guidelines should be dependant on the aim of the semantic network itself. The guidelines can also be described as restrictions and implemented into a software tool used for editing of semantic data.

2.3 Quality Assurance

Ideally, any new data in the network should be reviewed independently. As we have seen, there are plenty of ways how to import erroneous data into semantic networks. It may appear as self-evident but quality of semantic data is directly related to success rate of any NLP experiment that employs it or its usefulness when used as another language resource for linguistic work. If no guidelines exist for given semantic network then quality assurance may result in ad hoc fixes or random tweaks because no one knows what aspects of development were important in the past or when they may change again. Thus the quality assurance basically means a check to what extent the semantic data conform the guidelines. In that regard we can design and implement a set of automatic tests that would filter out lists of potentially erroneous semantic units for inspection, as described in the next chapter.

3 Heuristic Tests

As shown above, contrary to our best intentions, many different errors and inconsistencies may appear in our semantic network over time. These errors may become relevant when we need to use the data for our NLP experiment but don't have time and resources to fix the data directly. One way to quickly analyze the data is to design and implement a set of heuristic tests. Each test should be a formalized pattern of an error that appears multiple times within the semantic network. For example, Czech orthography allows us to use two different suffixes in words ending with *-ism* (e.g. in albinism). We can use a suffix with *s* or *z* in it – both *albinizmus* and *albinismus* are correct word forms in Czech. However, it may be useful in more than one way to use only one suffix variant consistently. In this case the test is very simple, we choose the desired variant of the suffix and let the test search each lexeme in our semantic network for the other suffix variant. On the output we get a list of candidate semantic units for review. Again, in this case the next step is very simple as there's virtually no way we could get a false positive from this test in Czech. We can simply apply all the proposed changes into the semantic network source database in batch-mode and we are done.

Most of semantic networks continue to be edited even after the main development project has ended. Once a test is implemented it is useful to have it scheduled for regular runs after a certain period of time via *cron* tool or any other scheduler software. The results automatically reported via e-mail can also help to keep the integrity of the network up-to-date at all times. Let's take a look at several more useful tests:

- **Morphology tests** In this category of tests we check for typing errors or for incorrect word forms, lemmata of which belong to the network. As a requirement we need a spell checking tool and a dictionary for our language (e.g. *ispell* [6]) but for highly inflectional languages such as Czech and other Slavonic languages it is far more useful to employ a morphological analyzer that can generate and recognize any word forms belonging to the language. If we use the *expand* model or use other means to automatically add semantic units for subsequent translation, morphology test can also filter out the data for us that has not been translated yet.
- **Syntax tests** Especially if we don't or didn't use any formal guidelines, any type of unexpected data can get into our semantic units. Usually they are various notes from the editors or redundant characters left over from automatic imports from other language resources. A simple test for non-letter characters and for high word counts in lexeme records can discover potentially erroneous semantic units. The advantage of this test is that it is much cheaper to employ than to implement a full set of syntactic restrictions directly into the software editing tool that is used to work with the data.
- **Instance test** Many cases of semantic relation pair class-instance (e.g. sea-Aegean Sea) are often marked as simple cases of hyperonymy-hyponymy in many semantic networks. To remedy this only a simple test for capitalized lexemes in semantic units is required to filter out most cases of named entities which should have their relations to their superordinate semantic unit changed to *Instance*.
- **Orphan nodes** Each part of speech has one significant semantic relation that connects all semantic units of its kind. For instance it is hyperonymy-hyponymy pair for nouns. Sometimes when new data are added to the network by hand or automatically, some of them remain unconnected thus creating orphan nodes within the network. A simple test can discover these nodes by checking each semantic unit for that particular semantic relation. If higher rate of false positives is not a problem this test can be extended to other relations as well, even if they are not supposed to interconnect every semantic unit in given category of semantic data.

Apart from the tests above many other language-specific or general tests can be designed according to particular needs of each semantic network. It should always be quicker to implement a test if we can find a pattern in the data than to do a full revision in top-down or alphabetical order.

4 Further Work

Although the heuristic tests are often very simple and quick to implement they can only cover the surface errors and inconsistencies visible on first sight. They can also help us to find various structural defects in a network such as undesired multiple inheritance, unbalanced trees or high sense number count for a lexeme but cannot offer a solution for such problems. Our further work

will therefore be focused on more sophisticated methods that would allow us to tackle practical problems with ontologies, data density or domain subtrees in a semantic network.

5 Conclusion

We have discussed an issue how to create and maintain semantic data in a semantic network that would allow us to minimize the number of errors and inconsistencies on surface level of the network. We have introduced a method of simple heuristic tests that can be easily implemented and can help us to remove frequent errors in the data even when the network is still being in development and many editors may participate in it. Although the tests are not an universal remedy to all problems we can have with the semantic data their favorable cost-benefit ratio makes then a useful tool to keep the integrity of our data intact.

Acknowledgements This work has been partly supported by the Ministry of Education of CR within the Center of basic research LC536 and in the National Research Programme II project 2C06009.

References

1. Miller, G., Beckwith, R., Fellbaum, C., Gross, D., Miller, K., Tengi, R.: Five papers on WordNet. International Journal of Lexicography **3**(4) (1990) 235–312.
2. Vossen, P.: Eurowordnet a multilingual database with lexical semantic networks. Computational Linguistics **25**(4) (1999).
3. Tufis, D., Cristea, D., Stamou, S.: Balkanet: Aims, methods, results and perspectives. A general overview. Science and Technology **7**(1-2) (2004) 9–43.
4. Vossen, P.: Right or Wrong. Combining lexical resources in the EuroWordNet project. In: M. Gellerstam, J. Jarborg, S. Malmgren, K. Noren, L. Rogstrom, CR Papmehl, Proceedings of Euralex '96, Goetheborg, Citeseer (1996) 715–728.
5. Sojka, P., Pala, K., Smrž, P., Fellbaum, C., Vossen, P., (eds.): Proceedings of the Second International WordNet Conference—GWC 2004, Brno, Czech Republic, Masaryk University Brno, Czech Republic (2004).
6. Kuenning, G., Willisson, P., Buehring, W., Stevens, K.: International ispell. Webpage can be found at: `http://fmg-www.cs.ucla.edu/fmg-members/geoff/ispell.html`, visited on February 17th (2004).