Filling the ___-s in Finnish MWE lexicons

Frankie Robertson
University of Jyväskylä
frankie@robertson.name

Abstract

This paper describes the automatic construction of FinnMWE: a lexicon of Finnish Multi-Word Expressions (MWEs). In focus here are syntactic frames: verbal constructions with arguments in a particular morphological form. The verbal frames are automatically extracted from FinnWordNet and English Wiktionary. The resulting lexicon interoperates with dependency tree searching software so that instances can be quickly found within dependency treebanks. The extraction and enrichment process is explained in detail. The resulting resource is evaluated in terms of its coverage of different types of MWEs. It is also compared with and evaluated against Finnish PropBank.

1 Introduction

This paper describes the automatic construction of a lexicon of Finnish Multi-Word Expressions (MWEs) derived from data in English Wiktionary[1] and FinnWordNet (Lindén and Carlson, 2010). A specific issue which is pronounced in — but by no means unique to — Finnish is is that of government. Consider the following examples:

(1) a. *Minä pidä-n kaku-sta*
I hold-1SG cake-ELA

'I like cake'

b. *Minä rakasta-n kakku-a*
I love-1SG cake-PAR

'I love cake'

c. *Minä pidä-n kaku-n*
I hold-1SG cake-GEN

'I keep (the) cake'

Contrasting 1a & 1b, we see that a particular verb may dictate the case of its argument. Conversely, contrasting 1a & 1c, we see that different cases of an argument can alternate with different senses of the same verb.

The perspective taken here is that such governance restrictions can be treated as simply another type of multiword. One justification for this approach is to consider an English transliteration of 1a where the elative case ending is translated using the preposition "from", i.e. the literal "I hold from cake". If we consider a hypothetical dialect of English where this was synonymous with "I like cake", then we could conceive of "hold from" as a prepositional verbal multiword.

[1] https://en.wiktionary.org/

Joint Workshop on Multiword Expressions and Electronic Lexicons, pages 13–21
Barcelona, Spain (Online), December 13, 2020.

These types of multiwords can be presented to humans in multiple ways, for example "pitää ___-sta" (to like), given just there in the author's preferred form of a gapped multiword, would commonly be presented in one of two other forms in a typical dictionary of Finnish. The first is as part of a headwords, where gaps would instead be rendered with an inflected pronoun e.g. "pitää jostakin", (jostakin = something-ELA). Alternatively, the gap might be specified in a grammar notes next to a particular word sense, in which case the entry under the headword pitää corresponding to "to like" would have a note "~ + elative" where ~ indicates the headword. Given this information is already specified in dictionaries, the focus of this paper is upon extracting it, alongside other types of Finnish multiwords, and making them machine readable so as to interoperate with other resources and systems.

The type of specifications given alongside individual definitions on Wiktionary go beyond simply verb-predicate argument-case associations, and include other types of morphological valency information, as well as constituent words, syntactic valency information (e.g. transitive/intransitive), fine-grained POS information (e.g. auxiliary), and occasionally semantic valency information.

A complimentary view on these these lexical items is that they are dependency tree templates, since, excepting semantic valency information, all this information is available within a Universal Dependencies (de Marneffe et al., 2014) parse tree. This perspective makes the simplifying assumption that a verb's arguments are its descendants within a dependency tree, which is not always the case.

Beside these syntactic frames, and more straightforward multiwords, the resource also includes inflections as another form of non-lemma idiomatic construction. Of interest is whether an inflected form is given a definition. If it is, this is a reasonable indication of idiomatic usage.

2 Related work

Related to this work are verb oriented semantic valency, or predicate-argument structured, Lexical Knowledge Bases (LKBs) such as PropBank (Palmer et al., 2005) and FrameNet (Baker et al., 1998). For Finnish, in this category there are Finnish PropBank (Haverinen et al., 2015), FinnFrameNet (Lindén et al., 2017) and FinnTransFrameNet (Lindén et al., 2019). The verbal frames within these resources do not concern themselves with syntax or morphology and can to some extent be preserved across languages, and so information about language specific issues such as the case in which a nominal argument appears are only visible through corpora annotated with these schemes.

In contrast, VerbNet is a fairly language specific formalism. Its frames give a great deal of specific semantic information about verbs, and also include syntactic restrictions on the parts of speech of arguments. In the case of the Basque Verb Index (Estarrona et al., 2016), which was inspired by the VerbNet formalism, this includes case information on arguments. Outside of VerbNet inspired formalisms, but within the Uralic languages, Wiechetek (2018) created a resource for Northern Saami, with both morphological and semantic category restrictions upon arguments, allowing identification of situations when the incorrect case had been used within a grammar checking application.

The approach in this paper is novel in its selection of initial data and the extent to which it is exploited. English Wiktionary gives definitions for words from many languages in English. Focussing in on the definitions of Finnish words on English Wiktionary, it can be seen as a unidirectional Finnish-English bilingual dictionary. As such, while it is written for everyday usage by a general audience, it is directed somewhat towards second language learners. The level of grammatical detail is thus driven by this intended audience, rather than by a specific linguistic formalism. While previous work in automatic creation of LKBs from existing resources such as DBnary (Sérasset, 2015), ConceptNet (Speer et al., 2017), and BabelNet (Navigli and Ponzetto, 2012) have made heavy use of Wiktionary, for the most part, detailed information about grammatical constructions has been neglected.

3 Method

The overall pipeline of linguistic data resulting in the FinnMWE resource is shown in Figure 1. The processing is performed with Python. For accessing FinnWordNet, NLTK (Bird et al., 2009) is used, while Wiktionary data is processed from the raw MediaWiki XML dumps using mwparserfromhell[2].

[2] https://github.com/earwig/mwparserfromhell

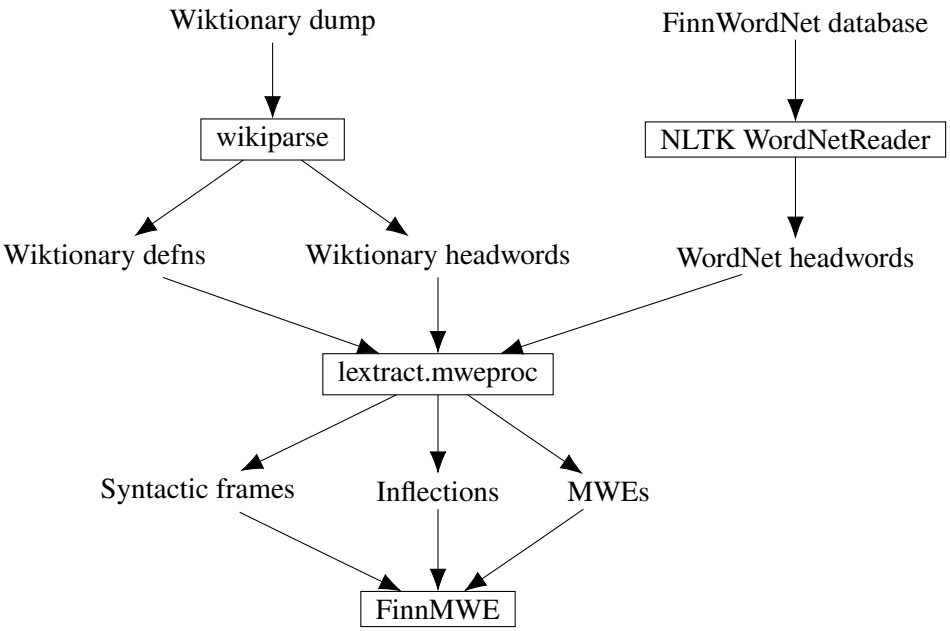

Figure 1: Diagram showing the data flow of linguistic data to create FinnMWE.

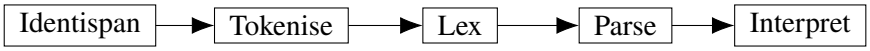

Figure 2: Pipeline to parse grammatical usage notes within Wiktionary definitions.

3.1 Sources

MWEs are obtained from two sources: FinnWordNet (Lindén and Carlson, 2010) and English language Wiktionary. Both Wiktionary and FinnWordNet contain data which can be used to create syntactic frames.

Within Wiktionary, there are multiple places MWEs can occur:

- The headword itself can be a multiword.
- The derived terms section of a page can contain MWEs expressed as headwords which are either links to other Wiktionary pages, or links which have not yet been created (known colloquially as redlinks).
- A word sense/definition entry within a page can contain a syntactic frame. On Wiktionary, the data is included within the text of a definition, for example, the headword pitää has the entry "(transitive + elative) To like, be fond of". In this case, the syntactic frame "pitää ___-sta" is extracted and associated with this definition.
  - The usage examples section can also contain MWEs, which can be extracted in a similar way to definitions.

MWEs in FinnWordNet are found only in headwords. When there is valency information in FinnWordNet, it is marked using abbreviated forms of pronouns. For example in the headword "pitää_kiinni_jstak" (hold onto something) jstak is short for jostakin (kiinni is a postposition with jostakin as its head), allowing the syntactic frame "pitää kiinni ___-sta" to be extracted.

Extraction of syntactic frames from collocation notes in Wiktionary word senses is more involved. The rule based information extraction pipeline is outlined in Figure 2. As the first step, spans which contain grammar notes are identified. These are typically visually separated from the definition text itself, e.g. by being bracketed. The main indicator that a bracketed part may contain a grammar note is the presence of certain words e.g. a case name "elative" or a tidle ~, which indicates the position of the headword.

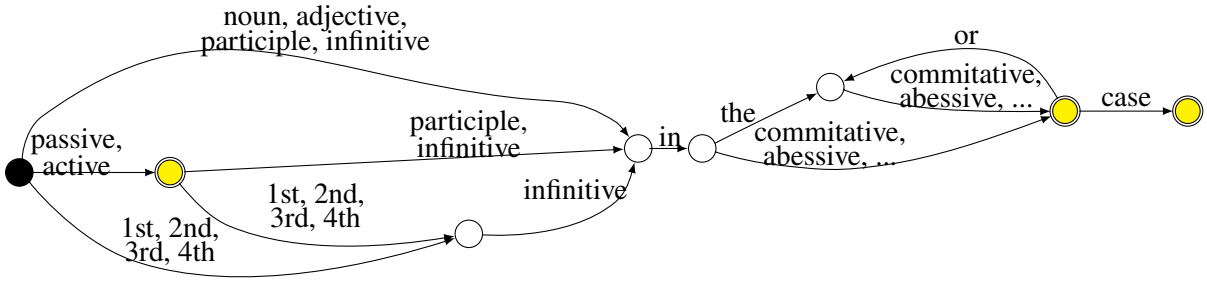

Figure 3: Fragment of a finite state automaton accepting grammar notes about Finnish nominal and nominalisation collocates. The black node is the starting state and the yellow nodes are accepting states.

Once a potential note is found, the lexing process maps surface tokens consisting of a mix of English words and MediaWiki markup to an intermediate set of normalised, type-tagged tokens. It is driven by an FST specified using the xfst language (Beesley and Karttunen, 2003) as implemented in HFST (Lindén et al., 2009). The input side of a fragment of the FST handling English language specifications of Finnish nominals and nominalised collocates is given in Figure 3.

The parsing step is implemented as a recursive descent Pratt (1973) parser. A Pratt parser extends the traditional recursive descent approach to context free parsing with a table-driven approach to operator precedence. For an example of where this is important consider the following headword-note pairs and their interpretation as bracket gapped MWEs.

yltää: intransitive + allative or illative ↔ yltää (___-lle OR ___-hin)

tulla: elative + 3rd person singular + noun/adjective in nominative or partitive **or** personal + translative
↔ (___-sta tulee (___ OR ___-ta)) OR (tulla ___-ksi)

In this case, it becomes apparent that *or* has a different precedence depending upon whether it is bold or not. The descending operator precedence order is: "/", "or", "+", "**or**", ";".

The interpretation step uses a cascade of heuristics to try and obtain MWEs from the final parse tree. The aim is to transform the tree into a state where it has a single root and consists only of plus-nodes and or-nodes, and finally to ensure that some node is marked as being the headword.

1. Merge directly adjacent (not separated by +) word features into word units.
2. Find or create a root, typically consisting of a plus-node, and abort if there is not exactly one.
3. Find all features outside the root, and merge them.
4. Ensure there is a word corresponding to the headword within each plus-node:
   (a) Features derived from certain strings such as "3rd pers. singular" and "personal" are always chosen as the headword node, even in preference to ~ (this is because sometimes ~ is misused as a generic blank).
   (b) Otherwise if there is a ~, which indicates the headword.
   (c) Otherwise if there is only one word, and it has the same part of speech as the head, assume it is the head.
   (d) Otherwise if there is any place where an empty node has been created in the parse tree, such as when there is nothing present on one side of a binary operator such as "+ elative" or "elative +" then pick one of these as the head node.
      i. If the first node in a plus-node is an empty node, pick this.
      ii. Otherwise if the last node in a plus-node is an empty node, pick this.
      iii. Otherwise just pick the first empty node left to right.

(e) Otherwise insert a new node as the headword at the beginning of the plus-node.

5. Merge all the merged features outside the root with the headword.

3.2 Finding the head

Finding which part of the MWE is the head can be helpful for identifying it in dependency trees, since if we make the argument constituency assumption, it will be at the root of the tree containing its arguments. For a Wiktionary definition or a usage example, it is clearly the case that the head is the same as the head of the title of the Wiktionary page. For a MWE Wiktionary headword, the head is sometimes explicitly specified in the etymology section, e.g. it may be formatted bold. Failing this, if the MWE occurs within the derived terms section of another page, we can assume that the head of the title of this page is its head.

For the remaining title derived Wiktionary definitions, the head must be guessed. This is, however, always necessary for FinnWordNet. In both cases, the guessing is done with the same procedure, based on the head and the MWE having the same part of speech, shown in detail in Algorithm 1.

Function FIND-HEAD(multi-word expression mwe from LKB lkb)
returns head h or **fails**
  $cand :=$ empty list
  for constituent word w in mwe **do**
    if w is a surface word **then**
      $w_{pos} :=$ all parts of speech of w in lkb
      if $|w_{pos} \cap mwe_{pos}| > 0$ **then**
        push w onto $cand$
      end
    end
  end
  if $|cand| = 1$ **then**
    return $cand_0$
  else
    fail
  end
end

Algorithm 1: The Find-Head procedure to find the head of an MWE.

3.3 Normalisation

As a normalisation step, all morphological information is converted into Universal Dependency features. For valency information, this means all information about case, infinitive, participles and so on are converted from the grammar usage note descriptions on the Wiktionary pages or the case abbreviation in a FinnWordNet headword into features on the consistent word. For part of speech tags, this means conversion from Wiktionary and WordNet part of speech to Universal Dependencies part of speech.

3.4 Storage, formatting and identification within text

Next, the normalised MWEs are saved as an SQLite database as an intermediate format for downstream applications. There are a series of formatters which directly make use of the collection of MWEs. The human readable formatters produce either a gapped MWE or one using pronoun abbreviations such as jstk. as in many Finnish dictionaries. In both cases, this is done by mapping from Universal Dependency features to normalised surface morphemes.

Another formatter exists for the purposes of creating search queries for SETS dependency tree search engine (Luotolahti et al., 2015). Since this is also based on universal dependencies, the mapping is mostly straightforward. However, one minor obstacle is Finnish's marginal accusative case. In Finnish this case only has a unique realisation for pronouns e.g. minä → minut, for other words it is realised as genitive e.g. kakku → kakun. This means that within corpora, the accusative is only annotated for

	Number	Proportion
Total multiwords	218 807	
. . . of which are syntactic frames within Wiktionary definitions	7726	3.5 %
. . . of which are extracted from Wiktionary titles	97 007	44.3 %
. . . of which are inflections	93 173	96.0 %
. . . of which are from a page without definitions	62 283	66.8 %
. . . of which are from a page with definitions	30 890	33.2 %
. . . of which are multiwords	3834	4.0 %
. . . of which are are a redlink	183	4.8 %
. . . of which are have a Wiktionary page	3651	95.2 %
. . . of which are extracted from FinnWordNet titles	114 074	52.1 %
. . . of which are syntactic frames	348	0.3 %
. . . of which are inflections	56 068	49.2 %
. . . of which are multiwords	57 658	50.5 %

Table 1: Table summarising contents of FinnMWE.

pronouns. Thus we map the accusative case within MWEs to the SETS dependency search language string `(PRON&Case=Acc)|(!PRON&Case=Gen)`, that is to say either a pronoun in the accusative, or something other than a pronoun in the genitive.

The FinnMWE toolkit also contains tools for extracting matches in morphologically analysed text by assuming they are contiguous or directly from Universal Dependencies trees without requiring an indexing step.

4 Evaluation

Table 1 gives basic information about the number of different types of multiwords in FinnMWE. The breakdown shows specifically how many Wiktionary inflections contain sense data, indicating they may be some kind of idiomatic usage, as well as how many multiwords come from redlinks, indicating they can only be found in the derived terms area.

Table 2 shows the results of comparing syntactic Wiktionary derived frames and semantic Finnish PropBank frames in its accompanying corpus. Since for a given hit for a headword, multiple MWEs can match, we use the powerset construction to make a discrete probability distribution of independent events. This distribution is compared against the distribution of PropBank frames found in the PropBank corpus using the entropy (in bits), mutual information and normalised mutual information (equivalent to the V-measure) defined as:

$$H(X) = -\sum_i p_i \log_2(p_i), \quad \mathrm{MI}(X;Y) = \sum_{y\in\mathcal{Y}}\sum_{x\in\mathcal{X}} p_{(X,Y)}(x,y)\log_2\left(\frac{p_{(X,Y)}(x,y)}{p_X(x)\,p_Y(y)}\right),$$
$$\mathrm{NMI}(X;Y) = \frac{2\,\mathrm{MI}(X;Y)}{H(X)+H(Y)}$$

For headwords with high normalised mutual information, the syntactic frame information from Wiktionary and the semantic frames of Finnish PropBank co-alternate. This means that the syntactic frames under this headword correspond to different senses according to the held-out LKB of PropBank.

5 Conclusion

This paper has introduced a large MWE and syntactic construction resource for Finnish based on FinnWordNet and English Wiktionary. The full extraction and processing pipeline is made available under the Apache v2 license at `https://github.com/frankier/wikiparse` and

			Wiktionary			PropBank		Agreement	
Headword	Gloss	Freq	Frames	Combs	Entropy	Frames	Entropy	MI	NMI
kannattaa	to support	54	3	2	0.69	2	0.68	0.61	0.88
vastata	to answer	104	6	7	1.31	3	1.06	0.93	0.79
pitää	to hold	442	18	18	1.78	14	1.72	1.09	0.62
ottaa	to take	324	3	2	0.69	21	1.94	0.49	0.38
käydä	to visit	151	11	15	1.94	18	2.10	0.72	0.36
lisätä	to add	102	2	2	0.69	4	1.13	0.30	0.33
saada	to obtain	688	11	5	0.63	11	1.40	0.31	0.30
tulla	to come	63	7	4	0.91	19	1.08	0.30	0.30
koskea	to touch	245	4	4	0.29	2	0.14	0.06	0.29
päästä	to reach	155	6	7	1.42	5	0.31	0.18	0.21
seurata	to follow	61	4	4	0.86	2	0.63	0.16	0.21
näyttää	to show	81	2	2	0.68	3	0.71	0.14	0.20
katsoa	to look	158	5	9	1.55	3	0.98	0.19	0.15
voida	to be able to	825	3	2	0.01	2	0.08	0.01	0.15
tehdä	to make/do	602	16	3	0.34	9	0.94	0.09	0.14
mennä	to come	165	4	4	0.90	11	0.97	0.07	0.08
todeta	to state	103	2	2	0.22	2	0.65	0.03	0.07
kuulua	to belong	131	2	3	0.56	3	0.17	0.02	0.06
antaa	to give	351	13	2	0.04	5	0.45	0.01	0.06
lukea	to read	86	4	2	0.11	5	1.01	0.03	0.05
päättää	to device	95	2	2	0.69	2	0.26	0.02	0.05
laskea	to calculate	96	3	2	0.06	6	1.37	0.02	0.02
istua	to sit	54	2	2	0.31	2	0.31	0.01	0.02
olla	to be	7866	33	4	0.01	25	1.45	0.00	0.00

Table 2: Table comparing distributions of syntactic frames from Wiktionary with frames from Finnish PropBank in its accompanying annotated corpus. Headwords with less than 50 results are excluded.

https://github.com/frankier/lextract. The final SQLite database is available to browse online, as well as to download at https://github.com/frankier/finnmwe.

Currently, the most fragile part of the processing pipeline is the extraction of information given within the body of Wiktionary pages, in particular the syntactic frame data. The reason is that this information is given as free text, and is only as consistent as it is by-convention, and so a Wiktionary editor could decide to introduce new conventions at any time. Therefore, one reasonable direction is to introduce more structure upstream. On more the conservative side, the current conventions on Wiktionary could be encoded into official template tags. A longer term solution would be to make sure this type of data can be encoded within the lexicographical data section of Wikidata.

Acknowledgements

Thank you to the reviewers for their valuable comments.

References

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 86–90. Association for Computational Linguistics.

Kenneth R Beesley and Lauri Karttunen. 2003. *Finite-state morphology: Xerox tools and techniques*. Center for the Study of Language and Information.

S Bird, E Loper, and E Klein. 2009. *Natural language processing with Python*. O'Reilly media.

Marie-Catherine de Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. 2014. Universal stanford dependencies: A cross-linguistic typology. In *LREC*.

Ainara Estarrona, Izaskun Aldezabal, Arantza Díaz de Ilarraza, and María Jesús Aranzabe. 2016. A methodology for the semiautomatic annotation of epec-rolsem, a basque corpus labeled at predicate level following the propbank-verbnet model. *Digital Scholarship in the Humanities*, 31(3):470–492.

Katri Haverinen, Jenna Kanerva, Samuel Kohonen, Anna Missilä, Stina Ojala, Timo Viljanen, Veronika Laippala, and Filip Ginter. 2015. The finnish proposition bank. *Language Resources and Evaluation*, 49(4):907–926.

Krister Lindén, Heidi Haltia, Juha Luukkonen, Antti Olavi Laine, Henri Roivainen, and Niina Väisänen. 2017. Finnfn 1.0: The finnish frame semantic database. *Nordic Journal of Linguistics*, 40:287–311.

Krister Lindén, Heidi Haltia, Antti Laine, Juha Luukkonen, Jussi Piitulainen, and Niina Väisänen. 2019. Finntransframe: translating frames in the finnframenet project. *Language Resources and Evaluation*, 53:141–171.

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

Krister Lindén, Miikka Silfverberg, and Tommi Pirinen. 2009. Hfst tools for morphology—an efficient open-source package for construction of morphological analyzers. In *International Workshop on Systems and Frameworks for Computational Morphology*, pages 28—47.

Juhani Luotolahti, Jenna Kanerva, Sampo Pyysalo, and Filip Ginter. 2015. Sets: Scalable and efficient tree search in dependency graphs. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 51–55.

Roberto Navigli and Simone Paolo Ponzetto. 2012. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Vaughan R Pratt. 1973. Top down operator precedence. In *Proceedings of the 1st annual ACM SIGACT-SIGPLAN symposium on Principles of programming languages*, pages 41–51.

Gilles Sérasset. 2015. Dbnary: Wiktionary as a lemon-based multilingual lexical resource in rdf. *Semantic Web*, 6(4):355–361.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 4444–4451. AAAI Press.

Linda Wiechetek. 2018. *When grammar can't be trusted-Valency and semantic categories in North Sámi syntactic analysis and error detection*. Ph.D. thesis, University of Tromsø.