Towards a Dependency-based PropBank of General Finnish

Katri Haverinen,[1,2] *Veronika Laippala,*[3] *Samuel Kohonen,*[2] *Anna Missilä,*[2]
Jenna Nyblom,[2] *Stina Ojala,*[2] *Timo Viljanen,*[2]
Tapio Salakoski[1,2] *and Filip Ginter*[2]

(1) Turku Centre for Computer Science (TUCS), Turku, Finland
(2) Department of Information Technology, University of Turku, Finland
(3) Department of Languages and Translation Studies, University of Turku, Finland

first.last@utu.fi

ABSTRACT

In this work, we present the first results of a project aiming at a Finnish Proposition Bank, an annotated corpus of semantic roles. The annotation is based on an existing treebank of Finnish, the Turku Dependency Treebank, annotated using the well-known Stanford Dependency scheme. We describe the use of the dependency treebank for PropBanking purposes and show that both annotation layers present in the treebank are highly useful for the annotation of semantic roles. We also discuss the specific features of Finnish influencing the development of a PropBank as well as the methods employed in the annotation, and finally, we present preliminary evaluation of the annotation quality.

KEYWORDS: PropBank, Finnish, dependency.

1 Introduction

Semantic role labeling (*SRL*) is one of the fundamental tasks of natural language processing. In a sense, it continues from where syntactic parsing ends: it identifies the events and participants, such as agents and patients, present in a sentence, and therefore it is an essential step in automatically processing the sentence semantics. SRL can be applied in, for example, text generation, text understanding, machine translation and fact retrieval (Palmer et al., 2005).

There have been several different efforts to capture and annotate semantic roles, the best-known projects being FrameNet (Baker et al., 1998), VerbNet (Dang et al., 1998) and PropBank (Palmer et al., 2005), all built for the English language. Out of the three resources, FrameNet is the most fine-grained one, defining roles for specific classes of verbs, such as *Cook* and *Food* for verbs relating to cooking. PropBank, in contrast, uses very generic labels, and is the only one of the three intended for corpus annotation rather than as a lexical resource. VerbNet, in turn, is between FrameNet and PropBank in granularity, and somewhat like PropBank, has close ties to syntactic structure. For a more thorough comparison of the three schemes, see the overview by Palmer et al. (2010).

The PropBank scheme in particular has become popular for semantic role labeling resources: after the initial effort on English, PropBanks for different languages have emerged, including, among others, PropBanks for Chinese (Xue and Palmer, 2009), Arabic (Zaghouani et al., 2010), Hindi (Palmer et al., 2009) and Brazilian Portuguese (Duran and Aluísio, 2011). As a PropBank is intended for corpus annotation purposes, and as the annotation scheme is closely tied to syntax, PropBanks are annotated on top of existing treebanks.

For Finnish, a freely available general language treebank has recently become available (Haverinen et al., 2010b, 2011), but no corpus annotated for semantic roles exists in the general domain. Haverinen et al. (2010a) have previously made available a small-scale PropBank of clinical Finnish, and thus shown that in principle, the PropBank scheme is suitable for Finnish and combinable with the Stanford Dependency (SD) scheme (de Marneffe and Manning, 2008a,b), the annotation scheme of both the clinical treebank and the general language treebank of Haverinen et al.

In this work, we present the first results of a project that aims to create a general language PropBank for Finnish, built on top of the existing Turku Dependency Treebank. This paper describes the methodology used for constructing the PropBank in a dependency-based manner, as well as shows the utility of the two different annotation layers present in the treebank. We also discuss the ways in which the Finnish PropBank relates to the English PropBank, our efforts to provide links between the two resources and the specific features of the Finnish language that require attention in the annotation process. Finally, we discuss the employed annotation methods and present preliminary evaluation.

2 PropBank Terminology

The purpose of a *Proposition Bank* or *PropBank*, as originally developed for English by Palmer et al. (2005), is to provide running text annotation of *semantic roles*, that is, the participants of the events described. For instance, the participants may include an *agent* who actively causes the event, or a *patient*, someone to whom the event happens. As defining a single set of roles that would cover all possible predicates is difficult, the PropBank annotation scheme defines roles on a verb-by-verb basis. Each verb receives a number of *framesets*, which can be thought of as coarse-grained senses for the verb. Each frameset consists of a *roleset*, which is a set of

act.01: to play a role, to behave	act.02: to do something
arg0 Player	arg0 Actor
arg1 Role	arg1 Grounds for action

Figure 1: Two framesets for the verb *to act*. The frameset *act.01* is intended for usages such as *He acted as her trustee* and the frameset *act.02* for usages such as *He acted on the knowledge that she betrayed him*.

semantic roles associated with this sense of the verb, and in addition, a set of syntactic frames that describe the allowable syntactic variations.

The *roles* or *arguments* in each roleset are numbered from zero onwards. A verb can have up to six numbered arguments, although according to Palmer et al. most verbs have two to four. The arguments zero and one (Arg0 and Arg1) have specific, predefined meanings: Arg0 is reserved for *agents*, *causers* and *experiencers*, and Arg1 is used for *patients* and *themes*. The arguments Arg2 to Arg5 have no predefined meanings, but rather they are specified separately for each verb. The original PropBank project makes an effort, however, to keep also these arguments consistent within classes of verbs defined in VerbNet (Dang et al., 1998). Figure 1 illustrates two framesets for the English verb *to act*.

In addition to numbered arguments, the PropBank scheme defines so called *adjunct-like arguments* or *ArgMs*. These, unlike the numbered arguments, are not verb-specific, but rather can be applied to any verb. The original PropBank defines a set of 11 different ArgMs: *location (LOC)*, *extent (EXT)*, *discourse (DIS)*, *negation (NEG)*, *modal verb (MOD)*, *cause (CAU)*, *time (TMP)*, *purpose (PNC)*, *manner (MNR)*, *direction (DIR)* and *general purpose adverbial (ADV)*. The distinction between numbered arguments and ArgMs is made on the basis of frequency: roles that occur frequently with a particular verb sense are given numbered argument status, and less frequent roles are left as ArgMs.

PropBanks are constructed in a data-driven manner using an underlying treebank. For each different verb present in the corpus, the verb senses observed are assigned framesets in a process called *framing*, and after the framesets have been created, the occurrences in the treebank are annotated accordingly. For each verb occurrence, the annotator must select the correct frameset and mark the arguments as defined in this frameset as well as the ArgMs.

3 The Turku Dependency Treebank

This work builds on top of the previously established Turku Dependency Treebank (TDT) (Haverinen et al., 2010b, 2011), which consists of 204,399 tokens (15,126 sentences) from 10 different genres of written Finnish. The text sources of the treebank are the Finnish Wikipedia and Wikinews, popular blogs, a university online magazine, student magazines, the Finnish sections of the Europarl and JRC-Acquis corpora, a financial newspaper, grammar examples from a Finnish reference grammar and amateur fiction from various web-sources.

The syntax annotation scheme of the treebank is a Finnish-specific version of the well-known Stanford Dependency (SD) scheme (de Marneffe and Manning, 2008a,b). The SD scheme represents the syntactic structure of a sentence as a directed graph, where the nodes represent the words of the sentence and the edges represent pairwise dependencies between them. Each dependency has a direction, meaning that one of the two words connected is the *head* or *governor* and the other is the *dependent*. Each dependency also has a *type* or *label*, which describes the syntactic function of the dependent.

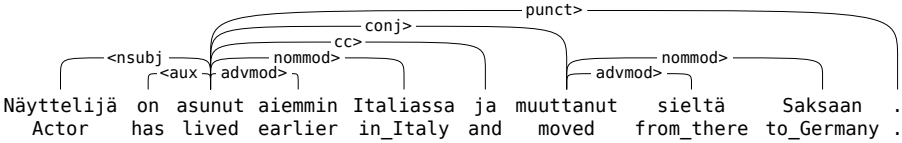

Figure 2: The SD scheme on a Finnish sentence. The example can be translated as *The actor has earlier lived in Italy, and moved from there to Germany*.

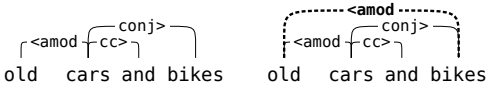

Figure 3: Conjunct propagation and coordination scope ambiguity. Left: the reading where only the cars are old. Right: The reading where both the cars and the bikes are old.

The original SD scheme contains 55 dependency types arranged in a hierarchy, where each type is a direct or indirect subtype of the most general dependency type *dependent* (*dep*). The scheme has four different *variants*, each using a different subset of the dependency types and giving a different amount of information on the sentence structure. The *basic* variant of the scheme restricts the sentence structures to trees, and the dependency types convey mostly syntactic information. The other variants add further dependencies on top of the tree structure, making the structures graphs rather than trees.

TDT uses a Finnish-specific version of the scheme, which defines a total of 53 dependency types and is described in detail in the annotation manual by Haverinen (2012). The annotation consists of two different layers of dependencies. The first annotation layer is grounded on the *basic* variant of the SD scheme, and hence the structures of the sentences in this layer are trees. The base layer of annotation is illustrated in Figure 2. The second annotation layer, termed *Conjunct propagation and additional dependencies*, adds on top of the first layer additional dependencies describing the following phenomena: *propagation of conjunct dependencies*, *external subjects* and *syntactic functions of relativizers*.

Conjunct propagation in the SD scheme provides further information on coordinations. The *basic* variant of the scheme considers the first coordinated element the head, and all other coordinated elements and the coordinating conjunction depend on it. Therefore, if a phrase modifies the first element of a coordination, it may in fact also modify all or some of the other conjuncts, and it should be *propagated* to those conjuncts that it modifies. Similarly, it is possible that all or some of the coordinated elements modify another sentence element. Conjunct propagation is used to resolve some (not all) *coordination scope ambiguities*; for instance, whether the adjective *old* modifies both *cars* and *bikes* or only *cars* in the phrase *old cars and bikes* (see Figure 3).

External subjects occur with *open clausal complements*, where a verb and its complement verb share a subject (*subject control*). The fact that the subject of the first verb is also the subject of the second verb cannot be marked in the first layer due to treeness restrictions, leaving it part of the second layer. *Relativizers*, or the phrases containing the relative word, such as *which* or *who*, are only marked as relativizers in the first layer of the treebank annotation, again in order to preserve the treeness of the structure. However, they also always have a secondary syntactic function, which in turn is annotated in the second layer of TDT. For instance, in *The man who stood at the door was tall*, the pronoun *who* acts as the subject of the verb *stood*. All of the phenomena addressed in the second layer of TDT are illustrated in Figure 4.

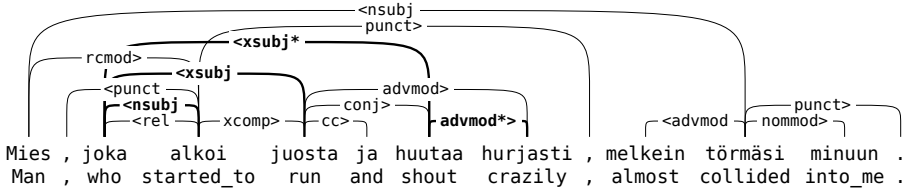

Figure 4: The second annotation layer of TDT. The example can be translated as *The man, who started to run and shout crazily, almost collided with me*. All second layer dependencies are in bold, and propagated dependencies are marked by an asterisk. The relative pronoun *joka* (*who*) also acts as the subject of the relative clause, as well an external subject to an open clausal complement. The external subject of the verb *juosta* (*run*) is also the external subject of the second coordinated verb, *huutaa* (*shout*) and is therefore propagated to the second conjunct. Similarly, the adverb modifier *hurjasti* (*crazily*) is shared between the two coordinated verbs. None of these phenomena can be accounted for in the first layer of annotation due to the treeness restriction.

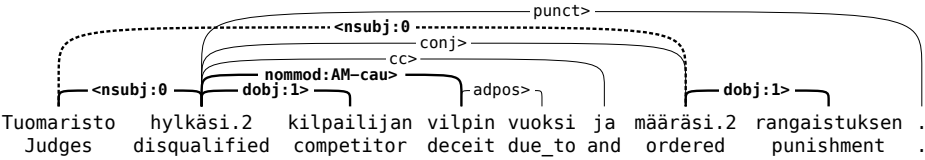

Figure 5: PropBank annotation on top of the dependency treebank. Dependencies with an associated PropBank argument are marked in bold. Note how one of the arguments (Arg0) of the latter verb in the sentence is associated with a second-layer dependency. The example sentence can be translated as *The judges disqualified the competitor due to deceit and ordered a punishment.*

4 Dependency-based PropBanking

The PropBank annotation of this work is built on top of the dependency syntax annotation of TDT, including both the first and second annotation layer. This is in contrast to the English PropBank, which has been built on top of the constituency-based Penn Treebank (Marcus et al., 1993). In the Finnish PropBank, each argument of a verb is associated with a dependency (be it first or second layer) in the underlying treebank, which means that the subtree of the dependent word, as defined by the dependencies of the first annotation layer, acts as the argument. For an illustration of the dependency-based PropBank annotation, see Figure 5.

In contrast to the original PropBank (Palmer et al., 2005) where in theory any constituent could be an argument, we make use of a heuristic: in most cases, the arguments of a verb will be its direct dependents. However, unlike the clinical language pilot study of Haverinen et al. (2010a), we do annotate all arguments, whether direct dependents of the verb or not. The heuristic of direct dependents being the likeliest arguments is only used to increase the speed of annotation by highlighting likely argument candidates for the annotator in the annotation software. In cases where an argument is found outside the dependents of the verb, we allow an extra dependency of the type *xarg* (*external argument*) to be added to any non-dependent word at annotation time, so that the argument can be attached to this dependency. For an illustration of external arguments, see Figure 6.

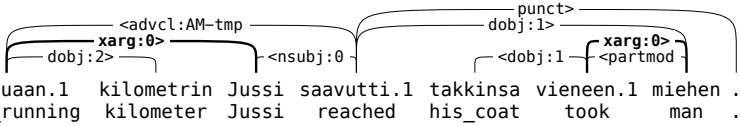

Figure 6: Arguments that are not direct dependents of the verb. On the left, the third person singular possessive suffix of the verb *juostuaan* (*after running, after he ran*) shows that it shares the subject of *saavutti* (*reached*), although this is not marked in the syntax annotation as the structure is not a case of subject control. On the right, semantically the noun *miehen* (*man*) is an argument to the verb *vieneen* (*took*), although syntactically, the verb participle modifies the noun. Note how by the assumption of whole subtrees forming arguments, the verb *vieneen* itself is incorrectly included in its own argument (Arg0) in the rightmost case. The example can be translated as *After running a kilometer, Jussi reached the man who took his coat.*

In the currently complete portion of the PropBank, 81.0% of all arguments, including both numbered arguments and ArgMs, are associated with a dependency of the first syntactic layer. If one takes into account dependencies of the second layer as well as the first, 93.1% of the arguments are covered, leaving a portion of 6.9% as external arguments. This shows that while the first layer of annotation does not suffice to cover an adequate proportion of the arguments, the second layer, which was annotated exactly for the purpose of finding semantic arguments falling outside the base-syntactic dependents of a verb, covers the majority of the remaining arguments.

As Choi and Palmer (2010) have shown, when using a dependency treebank for constructing a PropBank, in some cases the assumption that arguments are the dependents of the verb and their full subtrees results in some missing arguments that are directly due to the dependency structure of the sentence, as well as incorrect argument boundaries. In our work, the missing arguments are remedied by the *xarg* strategy, for instance in the case of a participal modifier, which is syntactically a dependent of the noun, although in fact the noun is its semantic argument. This is illustrated in Figure 6. In the case of a participal modifier, however, the addition of an *xarg* dependency leads to an incorrect argument boundary, as by the full subtree assumption the verb itself becomes part of its own argument. It should be noted that using the SD scheme already prevents some of the boundary issues mentioned by Choi and Palmer. For instance, in their work, modal verbs are problematic, as they are marked as the head of the main verb, whereas in the PropBank, the modal verb should be marked as an *ArgM-mod* for the main verb. In SD, however, the main verb is made the head and the auxiliary depends on it, which is unproblematic for PropBank annotation. A principled solution for the remaining boundary issues is not proposed in this paper, but is left as future work — perhaps using a rule-based approach, seeing that the boundary issues consist mostly of clear, regular cases.

5 Specific Features of Finnish Verbs

In the development of the Finnish PropBank, we have followed the same design principles as were used in the original PropBank: the arguments are numbered similarly from zero onwards, and the principles on which the framesets are created and distinguished are the same. We also use the same set of adjunct-like arguments, *ArgMs*, only adding two new subtypes, *consequence* (*CSQ*) and *phrasal marker* (*PRT*).

In order to expand the application potential of the Finnish PropBank to multilingual settings,

erota.2: leave a job		**quit.01: leave a job**	
Arg0	Person quitting	Arg0	Person quitting
Arg1	Job or position	Arg1	Job or position

Figure 7: Finnish and English verbs with corresponding framesets. The Finnish verb *erota* can be translated as *to quit*, and the framesets of this verb sense define identical argument structures. Therefore, the Finnish frameset is assigned the English as its corresponding frameset.

we assign to the Finnish frameset a corresponding frameset from the English PropBank where possible. Naturally, not all Finnish framesets have a corresponding English frameset, due to differences between the two languages. In this section, we discuss the specific features of the Finnish language influencing the creation of a PropBank, as well as the assignment of a corresponding English frameset and cases where no such frameset exists.

5.1 Frameset Correspondences and Non-correspondences

A frameset is assigned a corresponding English PropBank frameset when two conditions apply. The English verb must be a valid translation for the sense of the Finnish verb under consideration, and the two framesets must have the same arguments present, with matching argument numbers as well as argument descriptions. Occasionally, the argument descriptions of a corresponding English frameset are slightly rephrased in order to maximize the internal consistency of the Finnish PropBank.

As an example of corresponding framesets, one of the senses of the Finnish verb *erota* can be translated as *to quit* and it is used in contexts such as quitting a job or a position. This sense of the verb has its own frameset in the Finnish PropBank, and it is assigned a corresponding frameset in the English PropBank. The two framesets are illustrated in Figure 7.

For some verbs, however, the specific features of Finnish and the usages of the verbs being different to English do not allow assigning corresponding framesets. For instance, the frameset for the Finnish verb *korjata* meaning *to fix* or *to repair*, corresponds to neither of the English framesets, which, in turn, are also different from each other. The framesets for the three verbs are illustrated in Figure 8.

The difference between the two English framesets lies in the Arg2 argument; *to fix* includes an argument described as *benefactive*, which is absent in the description of *to repair*. The Finnish frameset, in contrast, contains an Arg2 describing an *instrument*, which is absent in both of the English framesets. Therefore it cannot be assigned either of them as the corresponding frameset. The addition of the *instrument* argument was necessary, however, as it is frequently found in the instances of the verb in the underlying treebank.

The corpus-based development of the framesets implies, naturally, that the non-correspondence of framesets does not necessarily indicate a difference between the languages. As the framesets are based on the treebank texts, they do not reflect all possible meanings and argument structures that a verb can have. This means that a non-correspondence can be caused merely by the limited and possibly different topics and text sources of the underlying treebanks. For example, the non-correspondence of the verb *korjata* with its English equivalents may be, at least partly, caused by contextual differences in the treebank texts.

A clear example of contextual differences causing non-correspondence of framesets is the Finnish

fix.02: to repair	
arg0	Fixer
arg1	Thing fixed
arg2	Benefactive

korjata.1: to fix, to repair	
arg0	Entity repairing something
arg1	Entity repaired
arg2	Instrument, thing repaired with

repair.01: to restore after damage or injury	
arg0	Repairer, agent
arg1	Entity repaired

Figure 8: Framesets of Finnish and English verbs with the meaning *to repair*. The Finnish frameset contains an argument describing the instrument of fixing, which is not present in either of the English framesets. Note that also the two English framesets differ in that the frameset for *to fix* contains a benefactive argument, whereas the frameset for *to repair* does not.

run.02: walk quickly, course or contest	
arg0	Runner
arg1	Course, race, distance
arg2	Opponent

juosta.1: move rapidly on foot	
arg0	Creature running, agent
arg2	EXT, distance
arg3	Start point
arg4	End point

Figure 9: Framesets of Finnish and English verbs describing *running*, the rapid movement of an agent on foot. The English frameset describes running a competition or a course, as in *John ran a marathon*, and the Finnish frameset describes running from one location to another, as in *John ran from home to work*. The abbreviation *EXT* on the Finnish frameset refers to *extent*, which is one of the ArgM subtypes defined in the PropBank scheme.

verb *juosta* and its English counterpart, *to run*, both of which describe the rapid movement of an agent. In the underlying Finnish treebank, the majority of examples describe an agent running from one location to another. However, the English PropBank does not contain a frameset for such a use of the verb *to run*, but rather only a frameset describing running a competition, distance or course. This is presumably due to the Penn Treebank only containing such examples, as the English *to run* can perfectly well be used for describing movement between two locations (see for instance the Collins English dictionary (2009)). As the examples present in the Finnish treebank require a frameset whose equivalent does not exist in the English PropBank, the framesets for these two verbs are necessarily different, as illustrated in Figure 9.

5.2 Finnish Causative Verbs and Polysemous Verbs in English

In addition to verbs differing by virtue of different usages, a more systematic difference between English and Finnish verbs is caused by the verb derivation system in Finnish. In English, many verbs, especially those of movement, are polysemous and can be used in different syntactic configurations. These verbs, also termed *variable behavior verbs* (see the work of Levin and Hovav (1994) and Perlmutter (1978)), can take as their syntactic subject either an agent actively causing an event or a patient merely undergoing it. For instance, the verb *to move* can have both subject types, as in *I move* versus *The chair moves*. In addition, the verb can also be used transitively, as in *I move the chair*, where the agent causes the event undergone by the patient.

In contrast, Finnish expresses the transitive meaning using a separate verb, typically formed by adding a morpheme to the root verb. For example, from the verb *liikkua* (*to move*, intransitive), it is possible to derive *liikuttaa (to make something move)*, as illustrated in Figure 10. These *causative* verbs can be formed both from originally transitive, such as *syödä (to eat)*, the

(1)	*I move*	*Minä liikun*
(2)	*The chair moves*	*Tuoli liikkuu*
(3)	*I move the chair*	**Minä liikun tuolia*
		Minä liikutan tuolia

Figure 10: Verbs taking both agents and patients as subjects in English and in Finnish. In English, the verb *to move* has three different uses: two intransitive uses and one transitive, where the agent causes the event occurring to the patient. In Finnish, this last sense, the transitive one, is expressed by a causative verb derived from the root verb.

liikkua.1: to move, be moved		liikuttaa.1: to move something	
arg0	Entity moving actively	arg0	Entity moving arg1
arg1	Entity whose movement something causes **if not arg0**	arg1	Entity moved
arg2	Place	arg2	Place
move.01: change location			
arg0	Mover		
arg1	Moved		
arg2	Destination		

Figure 11: Framesets of Finnish and English verbs with the meaning *to move*. Top left: Finnish, intransitive verb that takes as its subject either an agent or a patient. Top right: Finnish, transitive causative verb for moving. Bottom left: English, transitive and intransitive uses. Despite appearances, the frameset *liikuttaa.1* does not correspond to the frameset *move.01*, as the English verb is allowed to take either an Arg0 or an Arg1 as its subject, whereas the Finnish verb is not.

causative being *syöttää* (*to feed*), and intransitive verbs, such as *nukkua (to sleep)*, where the causative is *nukuttaa* (*to make someone sleep*) (Hakulinen et al., 2004, §311). For causatives and causativity in general, see for instance the work of Shibatani (1976) and Itkonen (1996), and the introduction by Paulsen (2011).

For the English PropBank, all three usages of the verb *to move* can be defined by a single frameset that includes both argument zero and argument one. Depending on the arguments present in a sentence, one or both arguments can be annotated, as PropBank does not require that all arguments are present in all examples. The formulation of the Finnish framesets and the assignment of corresponding framesets is, however, more challenging.

Because of its specific argument structure, the frameset for the Finnish causative derivation *liikuttaa (to make something move)* cannot be assigned the English *to move* as its corresponding frameset; *to move* can take either an Arg0 or an Arg1 as its subject, while *liikuttaa* can not. Despite this, the verb can still have the same arguments as the English frameset. As the Finnish intransitive *liikkua (to move)* is able to take either an agent or a patient as a subject, we assign it a single frameset that contains both an Arg0 and an Arg1, and explicitly mark that these arguments are mutually exclusive, meaning that only one of them should be annotated in any given example. Figure 11 illustrates the framesets of the two Finnish verbs and for comparison, the English verb *to move*.

It is also possible, although less common, for a Finnish verb taking alternatively an agent or a patient as its subject to allow a transitive usage. An example of this is the verb *lentää* (*to fly*),

where the intransitive with an agent (*lintu lentää*, *the bird flies*), the intransitive with a patient (*lentokone lentää*, *the plane flies*) and the transitive (*pilotti lentää lentokonetta*, *the pilot flies the plane*) all use the same verb. These verbs are treated similarly to the original PropBank as they are not problematic in the same way as the verbs described above, but they are nevertheless marked as variable behavior verbs in the frameset.

6 Annotation Protocol

The annotation of the Finnish PropBank, similarly to the English one, consists of two main phases. In the first phase, each verb is given a number of framesets that describe the different senses of the verb as they occur in the underlying corpus, and in the second phase, all the occurrences of the verb are annotated according to the framesets given.

In order to recognize tokens that require PropBank annotation, we use the open source morphological analyzer OMorFi (Pirinen, 2008; Lindén et al., 2009), which gives each token all of its possible readings with no disambiguation between them. In order to ensure the annotation of all verbs in the treebank, all tokens that receive a verbal reading, or a reading indicating that the word can be a *minen*-derivation (resembles the English *ing*-participle), are selected for annotation. Calculated in this manner, the Turku Dependency Treebank contains 49,727 potential verb tokens that require annotation, and 2,946 possible different verb lemmas. At this stage, 335 lemmas have been fully annotated, resulting in a total of 9,051 annotated tokens. This means that with respect to lemmas, approximately 11.4% of the work has been completed, and with respect to tokens, the estimate is 18.2%. It should be noted that when advancing from the common verbs towards verbs with less occurrences, the annotation becomes gradually more laborious. As illustrated in Figure 12, the amount of verbs with a large amount of occurrences is fairly small as compared to the amount of verbs with only few occurrences. The framing and annotation in this project commenced not from the most common verbs but rather those with a middle range occurrence rate, in order to settle the annotation scheme before moving to the most common verbs. Thus at this stage, the verbs with the most occurrences are in fact not yet annotated.

In total six different annotators, with different amounts of previous experience and different backgrounds, contribute to the PropBank, and the same annotators also act as framers. The verbs present in the treebank are framed and annotated one lemma at a time. In the beginning of the annotation process, all occurrences of each lemma were double annotated, in order to ensure high annotation quality even in the beginning phases of the project. As the work has progressed, we have gradually moved towards single annotation; high-frequency lemmas are partially double annotated, while low-frequency lemmas are single annotated. This is to increase the annotation speed while still being able to measure and control the annotation quality even after the initial learning phase.

In the case of double annotation, the two annotators assigned to a lemma create the framesets jointly, after which both of them independently annotate all occurrences using these framesets. At this stage, the annotator is required to mark both the numbered arguments and the adjunct-like arguments present in each example. Afterwards, the two analyses of each example are automatically merged, so that all disagreements can easily be seen, and in a meeting between all annotators, a single correct analysis is decided upon. Partially double annotated lemmas are framed in co-operation, and a portion of the occurrences is double annotated while the rest are divided between the annotators. In single annotation, each lemma is given to one annotator, and additionally, one annotator is assigned as a *consultant*, whom the annotator of the lemma

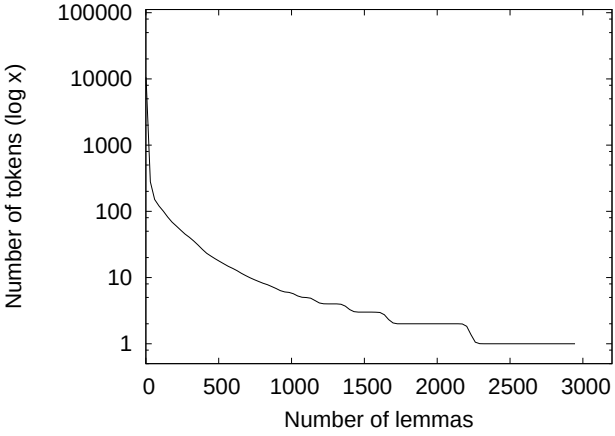

Figure 12: Numbers of verb lemmas of different frequencies as sorted from the highest number of occurrences to the lowest. High-frequency lemmas are relatively few, while many different low-frequency lemmas occur in the treebank text.

can turn to if facing problems with the framing. If unsure in the annotation phase, be it double or single annotation, an annotator can mark any argument as unsure. This function can also be used to signal suspected syntax-level errors in the treebank, as annotators are not allowed to alter the syntax at this stage.

In order to alleviate the labor-intensity of creating the framesets, *batches* of similar verbs are given their framesets simultaneously. When creating a new frameset for a lemma, the annotator is to consider whether there are other verbs that should also receive the same frameset, if such verbs are easily found. (The opposite is also possible: when considering a lemma, if the annotator finds that an existing frameset from another lemma can be re-used, they may copy the desired frameset for the verb under consideration.) For instance, if an annotator is considering the verb *to like*, possible other verbs that could receive the same frameset would be *to love*, *to care* or other verbs expressing affection that may have the same arguments. However, simply having the same arguments as in numbered arguments is not sufficient to be included in the same batch: for instance, verbs of dislike, although they also receive arguments describing the experiencer and the object of the feeling, should not be assigned to the same batch as the verbs of affection. In order to be included in the same batch, the verbs must have the same numbered arguments, and also the argument descriptions are required to be suitable for all verbs included.

This strategy has two benefits: in addition to saving time by creating framesets practically with no additional cost, it can enforce some consistency across the verbs. As a minor drawback, it requires additional care, as annotators should always make sure that the lemma they are considering does not already have the intended frameset as a side product of some other lemma. Also, if making a frameset for several verbs at once, care should be taken that verbs assigned simultaneously to other annotators do not receive framesets without these annotators' knowledge.

The distinctions between different framesets are made according to guidelines similar to those

used in the English PropBank, that is, the verb senses that the framesets correspond to are fairly coarse-grained. The main criterion used is that if two potential framesets have the same arguments (including the descriptions), or the arguments of one are a subset of the arguments of the other, only one frameset should be created.

The annotation is done using a custom software (see Figure 13) that allows the annotator to select a lemma to be annotated and then displays each occurrence as a separate case. The annotator must first select the correct frameset for the occurrence under consideration, and then assign the numbered arguments and adjunct-like arguments. All dependents of the verb occurrence are highlighted as default options for arguments, except for certain dependency types, such as punctuation, which never act as arguments. In case a dependency does not correspond to an argument, it is possible to leave the dependency unmarked. In addition, it is possible to mark a sentence element not depending on the verb as an argument using the *external argument* dependency. In addition to choosing one of the framesets defined, it is also possible to take one of the following actions. First, the annotator can mark an occurrence as *not a verb*, where the token is not in fact a verb but rather another part-of-speech, despite having a verbal reading assigned by OMorFi. Second, similarly it is possible mark the token to have a *wrong lemma*, where the token is a verb, but not of the lemma currently under consideration. Third, it is possible to mark the occurrence as an *auxiliary*, as in the PropBank scheme auxiliaries do not receive framesets or arguments.

7 Evaluation

In order to evaluate the performance of the annotators, we measure their *annotator accuracy* against the *merged* annotation. The accuracy is calculated using F_1-score, which is defined as $F_1 = \frac{2PR}{P+R}$. *Precision* (P) is the percentage of arguments in the individual annotation that are also present in the merged annotation, and *recall* (R) the percentage of arguments in the merged annotation that are also present in the individual annotation. For an argument to be considered correct, both its dependent word (the head word is the verb and thus always correct) and the argument number or the ArgM type must be correct. If the frameset of a verb is incorrect, then all numbered arguments of this verb token are considered incorrect as well. An ArgM of the correct type is judged correct regardless of the frameset of the verb, as ArgMs are verb-independent. For comparison, we also calculate *inter-annotator agreement* using *Cohen's kappa*, defined as $\kappa = \frac{P(A)-P(E)}{1-P(E)}$, where *P(A)* is the observed agreement and *P(E)* is the agreement expected by chance.

The overall annotator accuracy on the whole task is 94.1%, and the overall inter-annotator agreement in Kappa is 89.7%. While the F_1-score measures the accuracy of an annotator against the merged gold-standard, the Kappa-score measures the agreement between the annotators. It should also be noted that as the Kappa-score can only reasonably be calculated for labeling tasks, the external arguments, that is, arguments that are not syntactically direct dependents of the verb, are only taken into account in the F_1-score and not in Kappa. The per-annotator accuracies in F_1-score are listed in Table 1. The Table lists both overall scores and scores on numbered arguments and adjunct-like arguments separately, as well as the external arguments. These results show that overall, the accuracy is high, and that the adjunct-like arguments are more difficult to annotate than the numbered arguments, which is an expected result based on the figures previously reported by Palmer et al. (2005). The external arguments also seem to be more difficult than the numbered arguments in general. Some annotators show a large difference between precision and recall on the external arguments, indicating that these

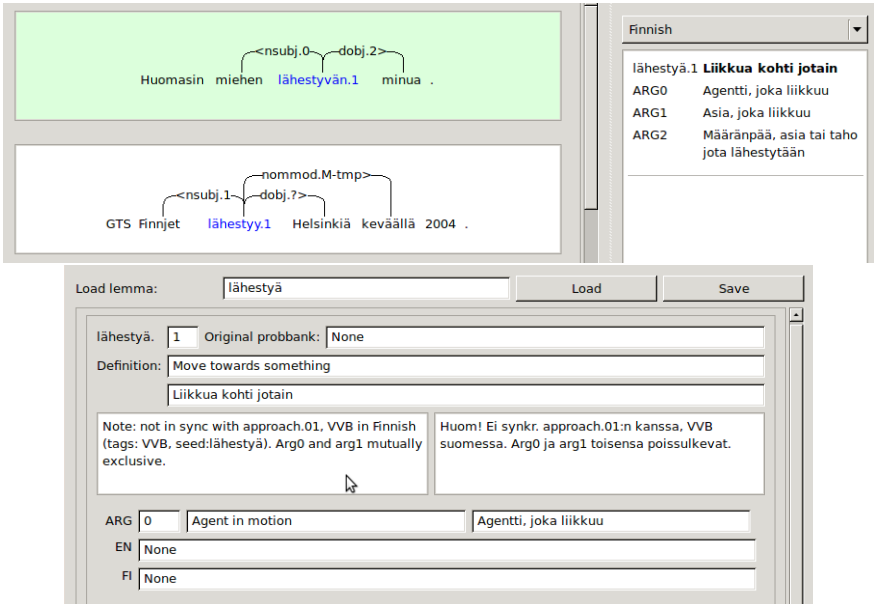

Figure 13: The annotation tool. Top: occurrence annotation. Two occurrences to be annotated are shown on the left, one marked as ready and one in mid-annotation. The direct dependents of the verb are shown as default alternatives for the arguments, and the question mark in the latter example indicates a dependency that has not been assigned with an argument. The framesets that have been created for the verb currently being annotated are shown on the right. Bottom: frameset editor. Each frameset has a number, a description, a field for the corresponding English PropBank frameset (not set in this example), as well as a free comment field. Similarly, each argument has a number, a description and a comment field. The comment fields may be used, for instance, for case requirements or use examples.

	Ann. 1	Ann. 2	Ann. 3	Ann. 4	Ann. 5	Ann. 6	All
Numbered (n=29,076)							
Recall	98.1	96.5	96.5	94.9	97.8	95.6	96.9
Precision	98.5	98.0	98.0	95.1	98.1	94.5	97.4
F-score	98.3	97.2	97.3	95.0	97.9	95.1	97.1
ArgM (n=15,771)							
Recall	92.5	86.6	87.3	83.7	90.1	82.8	87.8
Precision	92.9	87.3	86.6	85.2	92.6	82.0	88.2
F-score	92.7	86.9	87.0	84.4	91.3	82.4	88.0
xarg (n=3,118)							
Recall	93.3	80.8	79.3	70.3	87.4	85.9	86.0
Precision	97.8	97.8	92.3	70.3	94.7	84.3	92.7
F-score	95.5	88.5	85.3	70.3	90.9	85.1	89.2
overall (n=44,847)							
Recall	96.3	93.0	93.4	90.9	95.2	91.6	93.9
Precision	96.7	94.2	94.1	91.5	96.3	90.6	94.3
F-score	96.5	93.6	93.7	91.2	95.8	91.1	94.1

Table 1: Annotator accuracy results per annotator, both overall and separately for numbered arguments and ArgMs. Also a separate evaluation of the external arguments (*xarg*) is given. Note that for the F_1-scores the external arguments are also included in the counts of numbered arguments and ArgMs, seeing that each external argument is also one of these two argument types.

annotators forget to mark an external argument more often than mark an extraneous one. In addition to the possibility of overlooking an external argument, the task is made more difficult by the fact that with *xargs*, unlike the other arguments, the annotator is required to identify the correct token to act as the dependent.

Further, we evaluate the correctness of the frameset selections. Out of all frameset choices (including the possible choices of *not a verb*, *wrong lemma* and *auxiliary*), 88.4% were correct as measured against the final annotation result. Measured on only those instances where the frameset was correctly selected, the overall F_1-score was 94.6%.

8 Conclusions

In this work, we have presented the first results from a project aiming at a general Finnish PropBank. This PropBank is built on top of the previously existing Turku Dependency Treebank and utilizes both the first and second layers of syntax annotation present in the treebank, which are annotated according to the Stanford Dependency scheme.

We confirm the preliminary finding of the clinical language pilot study by Haverinen et al. (2010a) that the PropBank scheme can be used for Finnish and is compatible with the SD scheme. We also find that a large number of arguments are covered by the simplifying assumption that arguments are syntactic dependents of the verb; 81.0% of all arguments are accounted for when only considering the first layer of syntax annotation in TDT, and 93.1% if also the second layer is taken into consideration.

Regarding the quality of annotation, we find that the overall annotator accuracy of all six different annotators is 94.1% in F_1-score, and the accuracy on adjunct-like arguments (ArgMs) alone is 88.0%. The inter-annotator agreement in Cohen's kappa on the overall task disregarding

external arguments is 89.7%. From these figures we conclude that overall the quality of annotation is high, and that as expected, the adjunct-like arguments are more difficult to annotate than the numbered arguments. External arguments, with an overall F_1-score of 89.2%, are also more difficult than numbered arguments in general, due to the possibility of overlooking an external argument as well as the fact that for these arguments, the annotator also needs to identify the correct dependent word.

As future work, in addition to increasing the coverage of the PropBank, it would be beneficial to build rules to treat cases where the full subtree assumption of arguments fails, as well as enhance the annotation towards noun argument structures, that is, a NomBank (Meyers et al., 2004). The annotation could also be enhanced in several ways in order to accommodate, for instance, text generation, along the guidelines suggested by Wanner et al. (2012).

Acknowledgments

This work has been supported by the Academy of Finland and the Emil Aaltonen Foundation.

References

(2009). *Collins English Dictionary — 30th Anniversary Edition*. HarperCollins Publishers.

Baker, C. F., Fillmore, C. J., and Lowe, J. B. (1998). The Berkeley FrameNet project. In *Proceedings of COLING-ACL'98*, pages 86–90.

Choi, J. and Palmer, M. (2010). Retrieving correct semantic boundaries in dependency structure. In *Proceedings of LAW IV*, pages 91–99.

Dang, H. T., Kipper, K., Palmer, M., and Rosenzweig, J. (1998). Investigating regular sense extensions based on intersective Levin classes. In *Proceedings of COLING-ACL'98*, pages 293–299.

Duran, M. S. and Aluísio, S. M. (2011). Propbank-br: a Brazilian treebank annotated with semantic role labels. In *Proceedings of STIL'11*, pages 1862–1867.

Hakulinen, A., Vilkuna, M., Korhonen, R., Koivisto, V., Heinonen, T.-R., and Alho, I. (2004). *Iso suomen kielioppi / Grammar of Finnish*. Suomalaisen kirjallisuuden seura.

Haverinen, K. (2012). Syntax annotation guidelines for the Turku Dependency Treebank. Technical Report 1034, Turku Centre for Computer Science.

Haverinen, K., Ginter, F., Laippala, V., Kohonen, S., Viljanen, T., Nyblom, J., and Salakoski, T. (2011). A dependency-based analysis of treebank annotation errors. In *Proceedings of Depling'11*, pages 115–124.

Haverinen, K., Ginter, F., Laippala, V., Viljanen, T., and Salakoski, T. (2010a). Dependency-based propbanking of clinical Finnish. In *Proceedings of LAW IV*, pages 137–141.

Haverinen, K., Viljanen, T., Laippala, V., Kohonen, S., Ginter, F., and Salakoski, T. (2010b). Treebanking Finnish. In Dickinson, M., Müürisep, K., and Passarotti, M., editors, *Proceedings of The ninth International Workshop on Treebanks and Linguistic Theories (TLT9)*, pages 79–90.

Itkonen, E. (1996). *Maailman kielten erilaisuus ja samuus / Differences and Similarities of the World Languages*. Gaudeamus.

Levin, B. and Hovav, M. R. (1994). *Unaccusativity: At the syntax–lexical semantics interface*, volume 26 of *Linguistic Inquiry*. MIT Press.

Lindén, K., Silfverberg, M., and Pirinen, T. (2009). HFST tools for morphology — an efficient open-source package for construction of morphological analyzers. In *State of the Art in Computational Morphology*, volume 41 of *Communications in Computer and Information Science*, pages 28–47.

Marcus, M., Marcinkiwicz, M. A., and Santorini, B. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

de Marneffe, M.-C. and Manning, C. (2008a). Stanford typed dependencies manual. Technical report, Stanford University.

de Marneffe, M.-C. and Manning, C. (2008b). Stanford typed dependencies representation. In *Proceedings of COLING'08, Workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8.

Meyers, A., Reeves, R., Macleod, C., Szekely, R., Zielinska, V., Young, B., and Grishman, R. (2004). The NomBank project: An interim report. In *In Proceedings of the NAACL/HLT Workshop on Frontiers in Corpus Annotation*.

Palmer, M., Bhatt, R., Narasimhan, B., Rambow, O., Sharma, D. M., and Xia, F. (2009). Hindi syntax: annotating dependency, lexical predicate–argument structure, and phrase structure. In *Proceedings of ICON'09*.

Palmer, M., Gildea, D., and Kingsbury, P. (2005). The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Palmer, M., Gildea, D., and Xue, N. (2010). *Semantic Role Labeling*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Paulsen, G. (2011). *Causation and Dominance*. PhD thesis.

Perlmutter, D. (1978). Impersonal passives and the unaccusative hypothesis. In *Proceedings of the Fourth Annual Meeting of the Berkeley Linguistic Society*, pages 157–189.

Pirinen, T. (2008). Suomen kielen äärellistilainen automaattinen morfologinen jäsennin avoimen lähdekoodin resurssein. Master's thesis, University of Helsinki.

Shibatani, M., editor (1976). *The Grammar of Causative Constructions*, volume 6 of *Syntax and Semantics*. Seminar Press.

Wanner, L., Mille, S., and Bohnet, B. (2012). Towards a surface realization-oriented corpus annotation. In *Proceedings of INLG '12*, pages 22–30.

Xue, N. and Palmer, M. (2009). Adding semantic roles to the Chinese treebank. *Natural Language Engineering*, 15(Special issue 01):143–172.

Zaghouani, W., Diab, M., Mansouri, A., Pradhan, S., and Palmer, M. (2010). The revised Arabic PropBank. In *Proceedings of LAW IV*, pages 222–226.