# PARALLEL PROPOSITION BANK CONSTRUCTION FOR TURKISH

KORAY AK

IŞIK UNIVERSITY
2019

# PARALLEL PROPOSITION BANK CONSTRUCTION FOR TURKISH

KORAY AK

M.S., Computer Engineering, Işık University, 2011

Submitted to the Graduate School of Science and Engineering
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in
Computer Engineering

IŞIK UNIVERSITY
2019

IŞIK UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

# PARALLEL PROPOSITION BANK CONSTRUCTION FOR TURKISH

KORAY AK

APPROVED BY:

| | |
|---|---|
| Prof. Dr. Olcay Taner YILDIZ (Thesis Supervisor) | Işık University |
| Prof. Dr. Ercan SOLAK | Işık University |
| Assoc. Prof. Arzucan ÖZGÜR | Boğaziçi Univrsity |
| Prof. Dr. M. Oğuzhan KÜLEKÇİ | İstanbul T. Universıty |
| Dr. Nilgün Güler BAYAZIT | Yıldız Technical University |

APPROVAL DATE: .... / .... /....

# PARALLEL PROPOSITION BANK CONSTRUCTION FOR TURKISH

## Abstract

PropBank is the bank of propositions which contains hand-annotated corpus for predicate-argument information and semantic roles or arguments. It aims to provide an extensive dataset for enhancing NLP applications such as information retrieval, machine translation, information extraction, and question answering by adding a semantic information layer to the syntactic annotation. Via the added semantic layer, syntactic parser refinements can be achieved which increases the efficiency and improves application performance.

The aim of this thesis is to construct proposition bank for Turkish Language. Only preliminary studies were carried out in terms of Turkish PropBank. This study is one of the pioneers for the language. In this study, a hand annotated Turkish PropBank is constructed from the translation of the parallel English PropBank corpus, other PropBank studies for Turkish language examined and compared with the proposition bank constructed, automatic PropBank construction for Turkish from both parallel sentence trees and phrase sentences is analyzed and automatic proposition banks generated for Turkish.

**Keywords: Proposition bank, semantic role labeling, predicate-argument information, annotated corpus, lexical resources**

# PARALEL TÜRKÇE TÜMCE BANKASININ OLUŞTURULMASI

## Özet

PropBank yüklem-argüman bilgisi ve anlambilimsel rol ve argümanlar için el ile açıklanmış bütünceyi içeren bir tümce bankasıdır. Sözdizimsel açıklamaya anlambilimsel bir bilgi katmanı ekleyerek bilgi bulgetir, makine çevirisi, bilgi ayıklama ve soru cevaplama gibi doğal dil işleme uygulamalarını geliştirmek için kapsamlı bir veri kümesini sunmayı amaçlar. Eklenen bu anlambilimsel katman ile verimliliği arttıran ve uygulama performansını geliştiren sözdizimsel ayrıştırıcı geliştirmeleri elde edilebilir.

Bu çalışmada Türkçe tümce bankasının oluşturulması amaçlanmıştır. Bildiğimiz kadarıyla şu ana kadar Türkçe dilinde tümce bankası çalışması kapsamında birkaç çalışma yapılmıştır ve bu çalışma Türkçe dili için öncü niteliği taşıyacak bir çalışma olacaktır. Bu çalışmada el ile işaretlenmiş bir tümce bankası hazırlanmış, diğer çalışmalar incelenip, üretilen tümce bankası ile karşılaştırılmış, Türkçe için hem paralel cümle ağaçları kullanılarak hemde ağaç yapısında olmayan paralel cümleler ile otomatik tümce bankaları oluşturma incelenmiş ve otomatik tümce bankaları oluşturulmuştur.

**Anahtar kelimeler: Tümce bankası, anlambilimsel rol etiketleme, yüklem-argüman bilgisi, açıklanmış bütünce, sözcüksel kaynaklar**

## Acknowledgements

I am honored to present my special thanks and deepest gratitude to my supervisor Prof. Dr. Olcay Taner YILDIZ for his guidance in this thesis. Without his endless patience and support I would not finish this work. I am feeling lucky to share his vision and knowledge throughout this thesis.

I am also grateful to my family for their support. They have always been by my side whenever I needed.

*To my family. . .*

# Table of Contents

# List of Tables

# List of Figures

# List of Abbreviations

| | |
|---|---|
| **NLP** | Natural Language Processing |
| **SRL** | Semantic Role Labelling |
| **PropBank** | Proposition Bank |
| **WSJ** | Wall Street Journal |
| **POS** | Part-Of-Speech |
| **LVC** | Light Verb Construction |
| **AMR** | Abstract Meaning Representation |
| **APB** | Arabic Proposition Bank |
| **ATB** | Arabic Treebank |
| **MSA** | Modern Standard Arabic |
| **SVO** | Subject Verb Object |
| **VSO** | Verb Subject Object |
| **OSV** | Object Subject Verb |
| **OVS** | Object Verb Subject |
| **EADB** | Database for Basque Verbs |
| **TDT** | Turku Dependency Treebank |
| **SALSA** | Saarbrücken Lexical Semantics Annotation and Analysis |
| **HPB** | Hindi PropBank |
| **HDT** | Hindi Dependency Treebank |
| **PerPB** | Persian PropBank |
| **PerDT** | Persian Dependency TreeBank |
| **IMST** | ITU-METU-Sabancı Treebank |
| **UD** | Universal Dependencies |
| **TDK** | Turkish Language Association |

| | |
|---|---|
| **TNC** | Turkish National Corpus |
| **IBM** | International Business Machines |
| **EM** | Expectation Maximization |
| **FSM** | Finite State Machine |

# Chapter 1

# Introduction

## 1.1 Motivation

Semantic Role Labeling (SRL) is a well-defined task where objective is to analyze propositions expressed by the verb. In SRL each word bearing a semantic role in the sentence have to be identified. There are different types of arguments such as Agent, Patient, Instrument, and also adjuncts such as Locative, Temporal, Manner, Cause. These arguments and adjuncts represent entities participating in the event and give information about the event characteristics. As in the example below, "whisper" is the predicate which occurs between agent "the girl on the swing" and recipient "the boy beside her".

[The girl on the swing]$_{\text{Agent}}$ [whispered]$_{\text{Pred}}$ to [the boy beside her]$_{\text{Recipient}}$

Having semantic analysis of the corpora along with the syntactic architecture enhances NLP applications such as information retrieval, machine translation, information extraction, and question answering. Via the added semantic layer, syntactic parser refinements can be achieved which increases the efficiency and improves application performance. Over the past decade adding this semantic information layer to the syntactic annotation gathers more attention since improvements on machine learning algorithms and the developments of the computer systems enables and encourages processing large information resources. Prior to those developments researches were carried on with the help of manually created

semantic resources which is labor-intensitive and limited. After the development of statistical machine learning methods in the area of computational linguistics, learning complex linguistic knowledge became feasible for NLP applications.

The link between syntactic realization and semantic roles was mentioned in the comprehensive study of Levin [1]. In the study, syntactic frames were stated as a direct reflection of the underlying semantics. Syntactic frame sets are associated with Levin classes which defines the allowable arguments for each class member can take. VerbNet [2] extends classes defined by Levin. Abstract representation of syntactic frames for each class were added to the Levin Classes. These representations includes explicit correspondences between syntactic positions and the semantic roles. For example for "break" *Agent REL Patient*, or *Patient REL into pieces* added. FrameNet [3] is another semantic resource based on Frame Semantics theory. FrameNet proposes semantic frames to understand the meaning of most words which includes a description of a type of event, relation, or entity and the participants. For example, the concept of cooking typically involves an agent doing the cooking (Cook), the food that is to be cooked (Food), something to hold the food (Container) and a heat source. These objects are called frame elements. Words that evoke this frame are called lexical units of the frame. Other semantic resources specifically for SRL which provides input for developing statistical approaches are PropBank [4] [5], [6] [7] which includes predicate - argument structure by stating the roles each predicate can take along with the annotated corpora, and NomBank [8], an annotation project to list the arguments that are used with nouns in the PropBank Corpus like PropBank does for verbs. In both studies frame files were constructed to include possible arguments for the verb or noun. These frame files help the user to label the various arguments and adjuncts with roles.

Among these semantic resources PropBank is one of the important studies widely accepted by the computational linguistics communities. Penn Treebank Wall Street Journal [WSJ] news corpus has been annotated for semantic roles. These

argument roles are named and stored on framesets. Polysemous verbs have different framesets for different senses. PropBank uses same naming convention for Agent (Arg0) and Theme/Patient (Arg1) across verbs. Other set of arguments can vary from Arg2 to Arg5 which can represent different meaning across verbs. Since corpus is taken from WSJ, content is diverse and reliable and the collection itself has been a valuable language resource for linguistic research. PropBank has been applied for multiple different languages:

- Arabic: [9], [10]
- Basque: [11], [12], [13]
- Chinese: [47], [14]
- Dutch: [15]
- English: [4], [5], [6], [7]
- Finnish: [16]
- French: [17]
- German: [18], [19]
- Hindi: [20]
- Japanese: [21]
- Korean: [22], [23]
- Persian: [24]
- Portuguese: [25], [59]
- Spanish & Catalan: [26], [27]
- Turkish: [28], [29], [30], [31]
- Urdu: [32], [62]

In all those studies above, PropBank is used for understanding predicate argument structure of the language. To the best of our knowledge, only preliminary studies were carried out in terms of Turkish PropBank when we started working on this topic. Recently, Şahin presented their study [28] [29] for creating Turkish PropBank frames and Şahin and Adalı report their study [30] for the semantic role annotation of arguments in the Turkish dependency treebank.

## 1.2 The Contributions of this Dissertation

Although proposition banks are widely studied for different languages, Turkish lacks a comprehensive study and resources like annotated corpus and valency lexicon. The main contribution of this dissertation is the construction of such resources for Turkish language. The major contributions of the study is listed in order of appearance throughout the thesis.

1. A proposition bank for Turkish [31], is manually annotated using the translation of the English Penn-TreeBank. Framesets for 1914 verb senses are generated from 9560 sentences.

2. Proposition bank studies are explained in detail for 15 different languages apart from English and Turkish languages.

3. Recently published studies for Turkish PropBank ([28], [29], [30] and [31]) are compared in detail and proposition banks generated separately in these studies are mapped on frame level [33].

4. Automatic PropBank generation from English - Turkish parallel sentences is discussed. Sentences in tree structure and phrase structure are examined and annotation projection implemented with different methods.

## 1.3 Outline of Thesis

This thesis is organized as follows: In Chapter 2, we give some basic concepts and background information about the evaluation of semantic annotation and we state detailed information about Penn Treebank Project. Then we introduce English Proposition Bank and present PropBank studies in different languages. We also discuss Automatic PropBank generation by showing studies for automatic proposition bank generation in different languages. In Chapter 3, we provide information about Turkish Treebank which is the basis of the Turkish PropBank study and we present our annotation tool in detail for the work carried out for Turkish PropBank construction. In Chapter 4, we present the differences between constructed PropBank and PropBank study presented recently for Turkish. We compare both resources by matching verb senses and frame files. In Chapter 5, we present automatic Turkish proposition bank generation from English proposition bank using parallel sentence trees. We report the accuracy of generated proposition bank with respect to the hand annotated Turkish proposition bank. Then we propose our automatic proposition bank algorithms for parallel sentences not having tree structure. We show argument transfer from English PropBank using WordNet mapping and other word alignment techniques. In Chapter 6, we propose the future work, and finally we conclude the thesis.

# Chapter 2

# Literature Review

## 2.1 Penn Treebank Project

Penn Treebank [34] which is used as a basis for English PropBank, aims to provide an annotated corpus for investigators in different fields such as NLP, speech recognition, integrated spoken language systems and also theoretical linguistics. Penn Treebank corpus consists of over 4.5 million American English words. Corpus includes skeletal parses that covers syntactic and semantic information stored as linguistic trees. In the first years of the project (1989-1992), corpus is annotated with part-of-speech (POS) information and more than half of the corpus also annotated for skeletal syntactic structure. Corpus produced by the project is released through the Linguistic Data Consortium.

### 2.1.1 Pos Tagset

Different tagsets exist in the literature, most of them is fairly expensive. The POS tagset of Penn Treebank is based on Brown Corpus Tagset. Hovewer, number of tags is reduced from 87 to 48. 36 of those 48 tags are used for POS tags, remaining 12 tags are used for punctuation and currency symbols. Penn Treebank tagset is presented in Table 2.1.

Table 2.1: The Penn Treebank POS tagset : 36 tags are used for POS tags, remaining 12 tags are used for punctuation and currency symbols.

| | | | | | |
|---|---|---|---|---|---|
| 1. | CC | Coordinating conjunction | 26. | UH | Interjection |
| 2. | CD | Cardinal number | 27. | VB | Verb, base form |
| 3. | DT | Determiner | 28. | VBD | Verb, past tense |
| 4. | EX | Existential there | 29. | VBG | Verb, gerund/present participle |
| 5. | FW | Foreign word | | | |
| 6. | IN | Preposition/subordinating participle conjunction | 30. | VBN | Verb, past participle |
| | | | 31. | VBP | Verb, non-3rd ps. sing. present |
| 7. | JJ | Adjective | | | |
| 8. | JJR | Adjective, comparative | 32. | VBZ | Verb, 3rd ps. sing. present |
| 9. | JJS | Adjective, superlative | | | |
| 10. | LS | List item marker | 33. | WDT | wh-determiner |
| 11. | MD | Modal | 34. | WP | wh-pronoun |
| 12. | NN | Noun, singular or mass | 35. | WP$ | Possessive wh-pronoun |
| 13. | NNS | Noun, plural | 36. | WRB | wh-adverb |
| 14. | NNP | Proper noun, singular | 37. | # | Pound sign |
| 15. | NNPS | Proper noun, plural | 38. | $ | Dollar sign |
| 16. | PDT | Predeterminer | 39. | . | Sentence-final punctuation |
| 17. | POS | Possessive ending | | | |
| 18. | PRP | Personal pronoun | 40. | , | Comma |
| 19. | PP$ | Possessive pronoun | 41. | : | Colon, semi-colon |
| 20. | RB | Adverb | 42. | ( | Left bracket character |
| 21. | RBR | Adverb, comparative | 43. | ) | Right bracket character |
| 22. | RBS | Adverb, superlative | 44. | " | Straight double quote |
| 23. | RP | Particle | 45. | ' | Left open single quote |
| 24. | SYM | Symbol (mathematical or scientific) | 46. | " | Left open double quote |
| | | | 47. | ' | Right close single quote |
| 25. | TO | to | 48. | " | Right close double quote |

### 2.1.2 POS Tagging Process

POS tagging of Penn Treebank consists of two different phases. In the first phase, cascade of stochastic and rule-driven taggers makes automatic POS assignments with error rate 2-6%. Figure 2.1 shows an example sentence with automatic POS assignment.

Battle-tested/NNP industrial/JJ managers/NNS here/RB always/RB buck/VB up/IN nervous/JJ newcomers/NNS with/IN the/DT tale/NN of/IN the/DT first/JJ of/IN their/PP$ countrymen/NNS to/TO visit/VB Mexico/NNP ,/, a/DT boatload/NN of/IN samurai/NNS warriors/NNS blown/VBN ashore/RB 375/CD years/NNS ago/RB./.
"/"From/IN the/DT beginning/NN ,/, it/PRP took/VBD a/DT man/NN with/IN extraordinary/JJ qualities/NNS to/TO succeed/VB in/IN Mexico/NNP ,/, "/" says/VBZ Kimihide/NNP Takimura/NNP ,/, president/NN of/IN Mitsui/NNS group/NN 's/POS Kensetsu/NNP Engineering/NNP Inc./NNP unit/NN ./.

Figure 2.1: Sample POS tagged text - prior to correction phase

Then, a manual correction phase takes place. The automatic POS assignments, output of the first phase, is given to the annotators to be revised. Figure 2.2 shows an example sentence after correction phase.

Battle-tested/NNP*/JJ industrial/JJ managers/NNS here/RB always/RB buck/VB*/VBP up/IN*/RP nervous/JJ newcomers/NNS with/IN the/DT tale/NN of/IN the/DT first/JJ of/IN their/PP$ countrymen/NNS to/TO visit/VB Mexico/NNP ,/, a/DT boatload/NN of/IN samurai/NNS*/FW warriors/NNS blown/VBN ashore/RB 375/CD years/NNS ago/RB./.
"/"From/IN the/DT beginning/NN ,/, it/PRP took/VBD a/DT man/NN with/IN extraordinary/JJ qualities/NNS to/TO succeed/VB in/IN Mexico/NNP ,/, "/" says/VBZ Kimihide/NNP Takimura/NNP ,/, president/NN of/IN Mitsui/NNS*/NNP group/NN 's/POS Kensetsu/NNP Engineering/NNP Inc./NNP unit/NN ./.

Figure 2.2: Sample POS tagged text - After correction phase

### 2.1.3 Syntactic Tagset

Syntactic Tagset of Penn Treebank skeletal bracketing for phrases and null elements are presented in Table 2.2 and 2.3. Bracketing is again consists of two phases like POS tagging. In initial phase, a deterministic parser tries to parse the text. Then annotators corrects the output of the parser. Generally annotators do not reparse the text for bracketing instead they combine the partial bracketed output of the parser. Figure 2.3 shows a sample bracketed text.

Table 2.2: The Penn Treebank syntactic tagset for phrases.

| | | |
|---|---|---|
| 1. | ADJP | Adjective phrase |
| 2. | ADVP | Adverb phrase |
| 3. | NP | Noun phrase |
| 4. | PP | Prepositional phrase |
| 5. | S | Simple declarative clause |
| 6. | SBAR | Clause introduced by subordinating conjunction or 0 (see below) |
| 7. | SBARQ | Direct question introduced by wh-word or wh-phrase |
| 8. | SINV | Declarative sentence with subject-aux inversion |
| 9. | SQ | Subconstituent of SBARQ excluding wh-word or wh-phrase |
| 10. | VP | Verb phrase |
| 11. | WHADVP | wh-adverb phrase |
| 12. | WHNP | wh-noun phrase |
| 13. | WHPP | wh-prepositional phrase |
| 14. | X | Constituent of unknown or uncertain category |

Table 2.3: The Penn Treebank syntactic tagset for Null elements.

| | | |
|---|---|---|
| 1. | * | Understood subject of infinitive or imperative |
| 2. | 0 | Zero variant of that in subordinate clauses |
| 3. | T | Trace–marks position where moved wh-constituent is interpreted |
| 4. | NIL | Marks position where preposition is interpreted in pied-piping contexts |

```
( (S
    (NP Battle-tested industrial managers
        here)
     always
    (VP buck
        up
        (NP nervous newcomers)
        (PP with
            (NP the tale
                    (PP of
                    (NP (NP the
                            (ADJP first
                                    (PP of
                                        (NP their countrymen)))
                            (S (NP *)
                               to
                               (VP visit
                                    (NP Mexico))))
                        ,
                        (NP (NP a boatload
                                (PP of
                                    (NP (NP warriors)
                                        (VP-1 blown
                                            ashore
                                            (ADVP (NP 375 years)
                                                  ago)))))
                            (VP-1 *pseudo-attach*)))))))))
  .)
```

Figure 2.3: Sample bracketed text in Penn Treebank Project

## 2.2 English PropBank

PropBank is the bank of propositions where predicate-argument information of the corpora is annotated and semantic roles or arguments that each verb can take are posited. It is constituted on Penn Treebank Wall Street Journal [WSJ]. Primary goal is to label syntactic elements in a sentence with specific argument

roles to standardize labels for the similar arguments such as *the window* in *John broke the window* and *the window broke*. PropBank uses conceptual labels for arguments from Arg0 to Arg5. Only Arg0 and Arg1 indicate the same roles across different verbs where Arg0 means agent or causer and Arg1 is the patient or theme. The rest of the argument roles can vary across different verbs. They can be instrument, start point, end point, beneficiary, or attribute.

Moreover, PropBank uses ArgM's as modifier labels where the role is not specific to the verb group and generalizes over the corpora such as location, temporal, purpose, or cause etc. arguments. The first version of English PropBank, named as The Original PropBank, is constructed for only verbal predicates whereas the latest version includes all syntactic realizations of event and state semantics by focusing different expressions in form of nouns, adjectives and multi-word expressions to represent complete event relations within and across sentences.

### 2.2.1 The Original PropBank

In the first version of the English PropBank, annotation effort focused on the event relations expressed by only verbs. Prior to annotation, verbs of the corpora were analysed and frame files were created as stated in the Framing guidelines of PropBank [35]. Each verb has a frame file which contains arguments applicable to that verb. Frame files provide all possible semantic roles and also all possible syntactic constructions are represented with examples. Frame files may include more than one roleset with respect to the senses of the given verb. In the roleset of a verb sense, argument labels Arg0 to Arg5 are described with the meaning of the verb. For instance Figure 2.4 presents the roles of predicate "attack" from PropBank, Arg0 is "attacker", Arg1 is "entity attacked", and Arg2 is "attribute". In most of the rolesets two to four numbered roles exists. However, in some verb groups like verbs of motion there can be six numbered roles in the roleset. In the frame construction phase, numbered arguments are selected among the arguments and adjuncts in the sentence. Most of the linguists consider any argument higher

than Arg2 or Arg3 to be an adjunct. In PropBank, if any argument or adjunct occurs frequently enough with their respective verbs, or classes of verbs, they are assigned a numbered argument to ensure consistent annotation. Arg2 to Arg5 labels in the frame files may indicate different roles for the different senses of the verb.

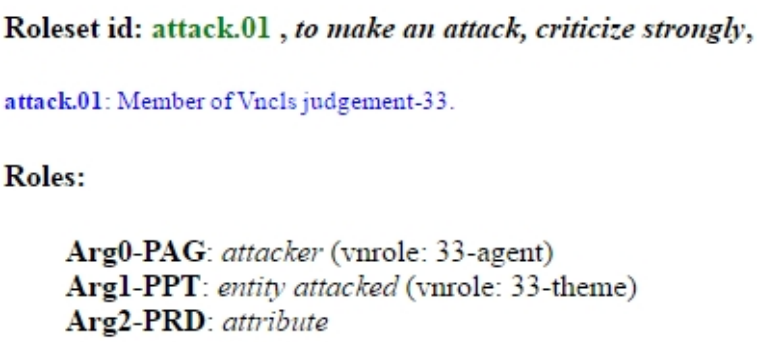


Figure 2.4: Roleset attack.01 from English PropBank for the verb "attack" which includes Arg0, Arg1 and Arg2 roles.

On the other hand, similar roles are assigned for Arg2 to Arg5 for the verbs in the same Levin class. Similar roles for the verbs in the same class are applied to some extent but, only a 50% of the verbs between VerbNet and Penn TreeBank II overlap. For example *buy*, *purchase* and *sell* are in the same class. The rolesets for *buy* and *purchase* are same and they are similar to *sell* rolesets since Arg0 role of the first group is equivalent to Arg2 role of the *sell* roleset. Below in Table 2.4 rolesets of these verbs represented.

Table 2.4: Rolesets *buy*, *purchase* and *sell* from English PropBank consist of the same roles.

| **PURCHASE** | **BUY** | **SELL** |
|---|---|---|
| ARG0: buyer | ARG0: buyer | ARG0: seller |
| ARG1: thing bought | ARG1: thing bought | ARG1: thing sold |
| ARG2: seller | ARG2: seller | ARG2: buyer |
| ARG3: price paid | ARG3: price paid | ARG3: price paid |
| ARG4: benefactive | ARG4: benefactive | ARG4: benefactive |

Verb types also affect the roles appear on the roleset. Unaccusative verbs like die, fall etc. are intransitive and have an experiencer as their subject. Although experiencer is the syntactic subject in the sentence, it is not a semantic agent. It does not actively initiate, or is not actively responsible for the action of the verb. Generally a telic and dynamic change of state or location is expressed by the unaccusative verbs while the opposite class, unergative verbs, tend to express an agentive activity. Also, inchoative senses of causative/inchoative verbs do not use a causing agent and demonstrate the situation as occurring spontaneously. Verbs like break, close, freeze, melt, open, can appear freely in both constructions. Figure 2.5 give examples for inchoative/causative constructions of the verb "'break". Some verbs like disappear do not allow causative, where some verbs like cut do not allow inchoative alternations. Inchoative/causative verb alternations are explained in detail with 31 verbs from 21 languages including Turkish in the study [36]. For the verb types that an agent can not participate, arguments start from Arg1.

(1) John broke the window (causative),(transitive)

(2) The window broke (inchoative),(intransitive)

Figure 2.5: Verb "break" can appear in both inchoative/causative constructions.

In the framing phase, whenever a frame created rolesets are copied to the other verbs in the same VerbNet class. Then generated frames are controlled. Generated rolesets seem suitable for almost all cases with a couple of exceptions. Also, rolesets for repeated and negated verbs are generated from the base verbs. While repeated verbs like re-enter shares same rolesets with the base verb enter, negated verbs created with un- prefix not always have same sense with the base verb even in some constructions base word and negated instance semantically same as in *ravel* and *unravel*.

PropBank also offers some special argument roles for different semantic and syntactic situations. A small set of verbs needs another argument to represent an

external force that cause semantic agent of the sentence to execute the action. As in the sentence "The general marched the soldiers to the tents" the semantic agent of the event march is *soldiers* but the marching event is caused by the *The general*. Here the external causer of the event is annotated with ArgA role since the causer induced the action. This tag is only used for verbs of volitional motion such as march and walk, modern uses of volunteer as in "Teacher volunteered the student sitting in front of the whiteboard to solve the question", graduate based on usages such as "Işık University graduates some of its students with honor degree".

Along with the semantic roles listed in the rolesets, verbs can take any of a set of general, adjunct-like arguments (ArgMs). Although verb-level negation and modal verbs are not considered adjuncts, they are also included inside the adjunct list. Also, discourse connectives is not adjunct, but included to ease future discourse connective annotation.

DIR: Directionals
LOC: Locatives
MNR: Manner
EXT: Extent
REC: Reciprocals
PRD: Secondary Predication
PNC: Purpose
CAU: Cause
DIS: Discourse
ADV: Adverbials
MOD: Modals
NEG: Negation

Figure 2.6: The list of ArgM's for tagging access modifiers of the predicate.

The list of ArgM's are presented in the Figure 2.6.

- Directional modifiers give information regarding the path of motion in the sentence. Directional modifiers may be mistakenly tagged as locatives.
- Locatives are used for the place that action take place.

- Manners define how the action is performed.
- Extent markers represent the amount of change occurs in the action.
- Temporal modifiers keep the time of the action.
- Reciprocals are reflexives which refer to other arguments like himself, itself, together, each other, both etc.
- Secondary predication markers are used for adjuncts of the predicate which holds predicate structure.
- Purpose clauses show the motivation for the action. Cause clauses are simply shows reason for an action.
- Discourse markers connect the sentence to the previous sentence such as also, however, as well, but etc.
- Adverbials used for syntactic elements that modify the sentence and are not labeled with one of the modifier tags stated above.
- Will, may, can, must, shall, might, should, could, would, and also going (to), have (to) and used (to) are modality adjunct of the predicate and are tagged with modal in PropBank.
- And finally negation is used to tag negative markers of the sentences.

Since roles are described with the meaning of the verb, annotators can easily understand and annotate the verb instances. After the frame files generated, corpora can be annotated with the given rolesets. In the annotation process, PropBank Annotation Guideline [37] can be used as an instructive source, since it includes several examples. As instructed in the guideline, annotation should begin with the numbered arguments first, then modifiers of the verb (ArgM's) should be annotated. In Figure 2.7, sample annotation is given where numbered arguments are listed first and then ArgM's are stated.

*Mr. Bush met him privately, in the White House, on Thursday.*

**Rel:** met
**Arg0:** Mr. Bush
**Arg1:** him
**ArgM-MNR:** privately
**ArgM-LOC:** in the White House
**ArgM-TMP:** on Thursday

Figure 2.7: Sample annotation sequence; annotation starts with numbered arguments and continues with access modifiers.

Semantic role annotation begins with a rule-based automatic tagger, and afterwards the output is hand-corrected. Annotation process is straight-forward, whenever a sentence is annotated annotator selects the suitable frameset with respect to the predicate and then tags the sentence with the arguments provided in the frameset file. Syntactic alternations which preserve verb meanings, such as causative/inchoative or object deletion are considered to be one frameset only. On the other hand, frame file may include different predicates such as keep has three different framesets in the frame file since the predicates "keep", "keep_up", and "keep_on" represent different semantic and syntactic behaviour, so annotator should select the correct frameset and tag the sentence with respect the roles provided in the frameset file. It is logical to start with Arg0 to the annotation since any argument satisfies two or more roles should be tagged with the highest ranked argument where the precedence is from Arg0 to Arg5.

Another issue in the annotation process is determining Arg0 and Arg1 in passive sentences. Subjects of the passive sentences are not the agent furthermore they are generally objects affected by the agent of the predicate so they are tagged with the Arg1 label.

PropBank offers solutions to annotation disagreements by adopting blind and double annotation to increase quality of the annotation. Whenever a disagreement occurs between the annotations, an adjudicator decides the correct annotation and new roles may be added to the roleset.

### 2.2.2 PropBank Today

Semantic information annotated in the first version of the PropBank is based solely on verbal predicates. Generally verbs provide majority of the event semantics of the sentence. However, to extract complete semantic relations of the event new predicate types such as nouns, adjectives, and complex predicate structures like light verbs should be taken into account. These new predicate types in the latest version of the PropBank offers guidelines specific to each structure that bears semantic information about the event.

Via different syntactic parts of speech, identical events can be expressed differently. Figure 2.8 gives examples for same events with different syntactic parts of speech. In the first example fear of mice is represented with verb, noun and adjective forms and gives the same semantic information about the event. The second example which represented with offer is also concludes the same semantic meaning across verb, noun and multi word constructions of the word. Semantic information is already covered for noun, adjective and complex predicates in FrameNet but PropBank recently expends its coverage for the new predicate types. For the nominal frame files PropBank relied on the NomBank in the initial creation of frames. Among all the noun types in NomBank, just the eventive nouns processed in PropBank. Also, WordNet and FrameNet are referred for expanding PropBanks nominal and adjective frame files coverage, and in assessing the derivational relationships between new predicate types rolesets.

| | | |
|---|---|---|
| She fears mice. | | He offered to buy a drink |
| Her fear of mice... | OR | His offer to buy a drink... |
| She is afraid of mice. | | He made an offer to buy a drink. |

Figure 2.8: Different syntactic constructions of the same event.

In the original PropBank adjectives followed by copular verbs as in the first example in Figure 2.8 are annotated with respect to the semantics of the copular verb. Annotation of the example sentence with respect to original PropBank

is shown in Figure 2.9. As you can see, annotation with respect to the verbal predicate in this sentence does not reveal the complete semantic meaning. A fearing event is not understandable from the annotation. The reason of incomplete semantic representation is the adjectives in these kind of sentences having more information semantically than the verbal predicates. To overcome this situation in the new version annotation has expanded to include predicate adjectives.

*She is afraid of mice.*

**Rel:** is
**Arg1-Topic:** She
**Arg2-Comment:** afraid of mice

Figure 2.9: Annotation of the sentence with respect to copular verb in Original PropBank.

The annotation of the same sentence with respect to predicate adjectives gives the result in Figure 2.10. Although the bulk semantic information is based on adjectives in this kind of sentences, the support verb does play a role in the sentence and annotation of the support verb is also required for complete semantic representation. The subject of the predicate adjective is syntactically an argument of the support verb be rather than the adjective afraid. To gather all the event participants in the sentence PropBank annotates support verb and its syntactic domain which contains the Experiencer argument. And then re-annotate sentence with respect to predicate adjective and its syntactic domain.

*She is afraid of mice.*

**Rel:** is afraid
**Arg0:** She
**Arg1:** of mice

Figure 2.10: Annotation of the sentence with respect to predicate adjective.

In the case of adjective predicate annotations, some constructions may occur along with the adjective. Gradable adjectives can carry their own semantics with

these constructions. Below example sentences with gradable adjectives are shown where adjective relations are in bold face.

1. We are too **hungry** to finish the work.
2. She is **clever** enough to solve complex problems.

Here the construction is termed as "Degree-Consequence" construction consists of a gradable adjective that is modified with some kind of degree word such as enough, too and an argument indicating a consequence for the degree. This construction can be used with any gradable adjective. However, some adjectives may not take any additional arguments. In the current guidelines this kind of arguments are tagged with "Construction" (CXN) tag to be processed in the future versions of the PropBank. So the example sentence is annotated as;

$$[\text{We}]_{\text{ARG0}}\ [\text{are too}]_{\text{ARG-CXN}}\ [\text{hungry}]_{\text{REL}}\ [\text{to finish the work}]_{\text{ARG-CXN}}.$$

Aforementioned these predicate types are already included in FrameNet. Adjectives are included in the frame elements for each semantic frame. As an example the adjective *sleepy* is one of the lexical units evoke *Biological urge* frame. It has participants such as *Experiencer* of the state which corresponds to Arg0 in PropBank. While the expansion of PropBank coverage with the new predicate types, an effort is spent to create new frames in accordance with both FrameNet and VerbNet. Each roleset created is mapped with VerbNet if it contains verbal predicates since VerbNet only includes verbs. And also, same frame is mapped to a FrameNet frame. VerbNet use a theta role, FrameNet is consists of frame elements where each role in the PropBank also mapped to these participants. This mapping, and an corpus annotated with the mapped roles, is an effort known as SemLink [38].

Also, PropBank recently added eventive and stative nouns which occurs inside or outside the light verb contructions to the focus of annotation. In the initial phase more than 2,800 noun rolesets are added to the frame files. Most of these rolesets

are taken from NomBank frames, and the coverage is expanded using WordNet definitions which states the noun types as noun.event, noun.act, noun.state for the eventive and stative nouns. Again as in the predicate adjectives, support verbs in the complex predicates such as the verbs in light verb constructions are annotated with their syntactic domains then annotation for the noun part is processed i.e. the light verb construction *make an offer* is annotated for both *make* and *offer*.

**ARG0:** entity offering
**ARG1:** commodity, thing offered
**ARG2:** price
**ARG3:** benefactive or entity offered to

[Yesterday]$_{\text{ARGM-TMP}}$, [John]$_{\text{ARG0}}$ [made]$_{\text{REL}}$ an [offer]$_{\text{REL}}$ [to buy the house]$_{\text{ARG1}}$ [for $350,000]$_{\text{ARG2}}$.

Figure 2.11: Annotation of the sentence with respect to noun in the LVC.

In the first version of the PropBank, nouns in the LVC's are ignored and the situation is handled either by using one of the rolesets of the dominant sense of the support verb or using a designated roleset for the LVC. As a result, semantic information presented by the noun is omitted. In the current version, annotators identify the light verbs and main noun predicate in the first pass then annotation is made with respect to complete arguments of the complex predicate by looking into the roleset of noun predicate. In the example in Figure 2.11, annotation is completed using the roleset of *offer*, and roles for both *made* and *offer* is extracted.

Expanding PropBank with new predicate types provide a broad coverage of event semantics as well as a robust interoperability with the Abstract Meaning Representation (AMR) project. In order to express event or state, AMR uses only a single roleset instead of having different rolesets for different syntactic realizations of the word as in PropBank. For this reason, while PropBank having multiple rolesets i.e. rolesets for fear-verb, fear-noun, and afraid-adjective, AMR generalize these syntactic variations to one roleset. Current PropBank is now unifying the rolesets across different parts of speech and having same meaning to better

**Event relation: Fear**

**Predicate:** fear-verb
**Roleset id:** fear.01 fear
**Roles:** ARG0: entity afraid
ARG1: afraid of what?
**Examples:** He fears bears.
He fears for his life.

**Predicate:** fear-noun
**Roleset id:** fear.01 fear
**Roles:** ARG0: entity afraid
ARG1: afraid of what?
ARG2: afraid for
**Examples:** His fear of bears.
His fear for his life.

**Predicate:** afraid-adjective
**Roleset id:** afraid.01 fear
**Roles:** ARG0: entity afraid
ARG1: afraid of what?
ARG2: afraid for
**Examples:** He is afraid of bears.
He is afraid for his life.

**Predicate:** fearful-adjective
**Roleset id:** fearful.01 afraid
**Roles:** ARG0: entity afraid
ARG1: afraid of what?
**Examples:** He is fearful of bears.
He is fearful for his life.

**UNIFIED ROLESET**

**Predicate**
**aliases:** fear, afraid, fearful
**Roleset id:** fear.01 fear
**Roles:** ARG0: entity afraid
ARG1: afraid of what?
ARG2: afraid for
**Examples:** He fears bears.

Figure 2.12: Rolesets of fear word prior to unification and aliases collected under the unified roleset.

**Event relation: Offer**

**Predicate:** offer-verb
**Roleset id:** offer.01 transaction
**Roles:** ARG0: entity offering
ARG1: commodity
ARG2: price
ARG3: benefactive or entity offered to
**Examples:** He offered to buy the house.

**Predicate:** offer-noun
**Roleset id:** offer.01 transaction
**Roles:** ARG0: entity offering
ARG1: commodity
ARG2: price
ARG3: benefactive or entity offered to
**Examples:** His offer to buy the house...
He made an offer to buy the house.

**UNIFIED ROLESET**

**Predicate**
**aliases:** offer-verb, offer-noun
**Roleset id:** offer.01 transaction
**Roles:** ARG0: entity offering
ARG1: commodity
ARG2: price
ARG3: benefactive or entity offered to
**Examples:** He offered to buy the house..
His offer to buy the house...
He made an offer to buy the house.

Figure 2.13: Rolesets of offer word prior to unification and aliases collected under the unified roleset.

comply with the use of rolesets from AMR project. A process named as 'aliasing' is used to aliase different lexical items of the same concept. As a result of this process fear-noun, fear-verb and afraid-adjective will be aliases associated with a single fear roleset. Similarly, offer-verb, offer-noun and the make offer-light verb construction will be aliases of a single offer roleset. Currently, there exists 7312 unified frame files in the github site of the project. Figure 2.12 and 2.13 shows the pre and post situation of rolesets after aliasing process.

As you noticed fear rolesets do not contain the same arguments. Fear-noun and afraid-adjective have Arg2 argument in the roleset whereas fear-verb and fearful-adjective do not have. In such situations the different arguments are either dropped or preserved and in case of any word is tagged with this argument is updated in the corpora. Mostly arguments are preserved if they contain an important information for the event and cannot be replaced with and modifier.

On the other hand, FrameNet handles these kind of lexical units differently. FrameNet list these derivationally related lexical units to different frames while PropBank and AMR groups them into the same frame or rolesets. For example FrameNet splits fear event relation of PropBank into two different frames. Fear-noun is included in the 'Fear' Frame and fear-verb is listed in the Experiencer Focus frame, and afraid-adjective added to the both of the frames. Fearful-adjective is not found in FrameNet at all.

## 2.3 PropBank Studies in Different Languages

As stated previously in Chapter 1 PropBank has been applied for multiple different languages. In this section, PropBank studies in different languages and the challenges to construct such dataset will be discussed in detail.

### 2.3.1 Arabic PropBank

Arabic Proposition Bank (APB) [9] and [10] is built on the Arabic Treebank (ATB) [39] and an Arabic morphological analyser, AraMorph [40]. Arabic is used by over 300 million people in the world. It is also read in different areas of the world since the holy book Quran is written in Arabic language. There exist several spoken vernaculars in the community. Proposition bank is built on formal written form of the language which is referred as modern standard Arabic (MSA).

Prior to the Arabic Proposition Bank construction, English, Chinese and German proposition banks were already constructed. However, Arabic language has

different morphologic and syntactic characteristics which affect the process of creating proposition bank for the language. It has templatic morphology, words are formed with roots and affixes, and clitics are attached to the words. Romanized arabic word "wbHsnAthm" which means "and by their virtues[fem.]" is given as an example in [9]. It is splitted as the conjunction w "and", preposition b "by", the stem HsnAt "virtues [fem.]", and possessive pronoun hm "their". Verbs of the language are also marked for tense, voice and person like English. However, verbs can also be marked with mood (subjunctive, indicative and jussive). Furthermore, nominals can be marked with case (accusative, genitive and nominative), number, gender and definiteness features.

Also, Arabic differs from the other languages since it is a pro-drop language. In Arabic, subject of the verb is dropped in some sentences. Instead the subject information is encoded in the verb. One-third of sentences in the Arabic Treebank are pro-dropped in terms of subject. Arabic also has relative free word order. It allows sentences like subject-verb-object (SVO) and verb-subject-object (VSO) argument orders, as well as, OSV and OVS. VSO and SVO ordered sentences are equally found in Arabic Treebank where each type has 35% occurences. Another difference is the expression of possession information in noun phrases. It is defined as idafa constructions where prepositional phrase is omitted in noun phrases where an indefinite noun is followed by a definite noun. "rjl Albyt" is given as an example in the paper which is translated as "man the-house" meaning "man of the house".

In the frame creation phase, the level of granularity for frame entries has several options for Arabic. Most of the verb roots in Arabic are triliteral. Verb lemmas are derived from the root with respect to the vocalic structure. In the Arabic writing system, it is not easily achievable deriving both the lemma and the root from the surface form of the word. So, lemmas are used instead of verb roots to index predicates in the propbank since finding roots from lemmas is not deterministic. Each unique sense has its own frameset. In the first version of the APB frame files are stored in hand written xml files whereas in the second version Jubilee and Cornestone [41], [42] are used for frame creation and annotation. 1955

frame files with 2446 framesets are constructed for the APB.

### 2.3.2 Basque PropBank

Basque PropBank [11], [12], [13] is built on syntactically annotated Basque Corpus (EPEC). This corpus contains 300,000 word sample collection of written standard Basque. About 100,000 of the words are gathered from Statistical Corpus of 20th Century Basque. The rest of the words are from a daily newspaper. The total number of verbs in the corpus is 622. This number is very small compared to the other languages because different verbs in other languages are represented with verb constructions having adverbial components. For example the English verb "to bike" is represented as *bizikletaz ibili* which means "to go on bike", likewise "to walk" is translated as *oinez ibili*, "to go on foot". Alongside the PropBank model, a database which includes syntactic/semantic subcategorization frames for Basque verbs (EADB) [43] and Basque dependency treebank [44] is used in the proposition bank construction.

In the beginning of the construction, 3 verbs: *esan, eskatu* and *adierazi* from the top 5 most occured verbs in EPEC corpus is chosen for the study [11]. In study [43], Aldezabal offers syntactic-semantic frames for each verb, this auxiliary information is used for selecting PropBank frames. By using the senses, number of argument and declension cases for each verb, corresponding PropBank frame is selected since similar syntactic-semantic behaviour is expected. After finding corresponding English words from the dictionaries, words having similar syntactic features are paired. In this preliminary study, a file for each verb is generated which contains all the sentences that include the verb. Then these sentences are tagged with respect to argument roles of the verb. After the tagging process, problematic cases are discussed, frame files are updated and annotations are reviewed if necessary.

Also, automatic tagging of the corpus is discussed by looking the similarities of the annotations for these 3 verbs. Some heuristics are explained for automatic

tagging, for all annotated instances, the ergative case takes ARG0 which means all ergatives can be tagged as ARG0. On the other hand, in the absolutive case tagging is more complex, in the two rolesets of esan, it takes ARG1 for esan.01, and ARG2 for esan.02. Completive and the instrumental cases are unambiguous so the instances of the verb can be annotated automatically. The instances having COMP or INS constituent can be disambiguated, which constitutes 80% of the occurrences by COMP and 3% by INS. And the remaining ~18% can be tagged by human taggers.

In the next study [12], Aldezabal announced the attempt for the annotation of 100 verbs from the Basque Dependency Treebank. An annotation tool, AbarHitz [45] is adapted for the annotation of semantic roles. Also, English-Basque verb mapping [46] is used for the identification of corresponding English verbs, it is based on Levin's alternations and classification. All the verbs in Levin [1] translated to Basque and a mapping [46] is constructed by looking semantic class information and syntactic similarity. The mapping is then used for matching predicates to English counterparts for acquiring semantic roles. However, since the mapping is outdated, some of the entries are changed or deleted from the Levin's classes. So only 57% of the entries are used for the mapping which covers 46% of the EPEC corpus. A semantic tag is defined as *arg_info* which contains the following fields;

- **VN (VerbNet/PropBank verb):** PropBank sense for the English verb.
- **V (Verb):** main verb of the sentence.
- **Treated Element (TE):** the word tagged with respect to predicate.
- **VAL (valence):** argument value of the word.
- **VNrol (role in VerbNet):** the role of the annotated argument in the sentence.
- **EADBrol:** semantic role according to EADB roleset

- **HM:** (Selectional Restriction). [+animate], [-animate], [+count], [- count], [+hum], [-hum]

In Figure 2.14, arginfo tag for the word "Argentinara" (to Argentina) with respect to verb "joan" (to go) is presented. These tags are then presented in the annotation tool AbarHitz, along with the information from PropBank/VerbNet. For some of the words these tags are automatically generated, but some of the words has incomplete since the tags are not determined automatically. With these pre-tagged annotations, annotation phase is facilitated, and 12,000 words of the corpus is annotated.

(3) Argentinara joan zen taldea egongo da Pau Orthezen kontra.
The team that went to Argentina will play against Pau Orthez
arg_info: (go_01, joan, Argentinara, Arg4, Destination, end_location, -4).

Figure 2.14: "Argentinara" is annotated for predicate-argument relation.

In [13], annotated word count is increased to 37,000 and 32 verbs in the corpus. These verbs are the most occured verbs which are also available in the EADB [43]. In the final study, complex cases for creating the Basque PropBank is discussed. One of them is to map Basque verbs to the English counterparts. Some verbs have single match, but some of them are matched more than one PropBank verb. In this case, syntactic features are compared and the most suitable verb is selected. Also, some verbs can not be matched with PropBank verbs, where different resources like Verb-Index (http://verbs.colorado.edu/verb-index/index.php) are used to make a mapping. Moreover, some verbs in Basque do not share the same arguments with the matching PropBank verb with respect to EADB database. For that case, a new sense for the verb is generated. Some matched verbs are not tagged with the roles from PropBank frames whenever the role is not clear in the sentence and the "-" symbol is used for the annotation. Adjunct tags are preferred instead of numbered arguments (ARG2-ARG5) in some situations. Also, if two different roles are assigned to the same word, the role in the EADB is used for the annotation.

### 2.3.3 Chinese PropBank

Chinese is one of the first languages that constructed proposition bank. Chinese PropBank [47], [14] is build on Chinese Treebank [48] which contains sentences from couple of news agencies from China, Sinorama magazine from Taiwan, broadcast conversation, news groups, and web log data. The corpus has more than one million words which are fully segmented, POS-tagged and annotated with phrase structures.

Prior to annotation, frame files are generated. Frame files contains semantic roles and also subcategorization frames that actualize those roles. An effective test for English verb classification, "diathesis alternations" by Levin [1] is also viable for Chinese and used to distinguish some of the verb senses. The assumption behind of the diathesis alternation is verbs having similar diathesis alternations patterns also share similar semantic information. Different senses of verbs detected by examining the divergent diathesis alternation patterns and framesets are created for those verb senses. As an example the Chinese verb "tongugo" has two senses; "pass a bill or law" and "pass through a tunnel". First sense of the verb allows "object of transitive/ subject of intransitive" alternation in which an argument syntactically in object position of the transitive sentence becomes subject of the sentence in intransitive version. Whereas the same alternation is not applicable to the second sense of the verb. Although diathesis alternation gives information about sense distinction, it does not distinguish all verb senses properly.

Some complex cases are discussed for sense distinction in the study. First of them is idioms and metaphors. Idioms are expressions where the verb is headed by the same word and the meaning is irrelevant to the meanings of the composing words. They generally require a distinct frameset and separated from the other senses of the verb. Metaphors are substitution of a thing for another in order to suggest comparison or resemblance, they generally have a similar verb usage and do not require a separate frameset. Light verbs also complicate the sense determination. Light verbs generally occur with a nominalized predicate.

In this structure the meaning is different from the non-light verb usage of the same verb and argument structure is mainly defined by the nominalized verb. Furthermore, In Chinese phrasal verbs are confusing for sense selection. There exist three categories for multi word compounds. First category contains multi word compounds which none of the words is the head as in example “kaifa/develop shengchan/produce”. This type of the verbs are annotated separately for each word in the phrase and no annotation is done with respect to whole compound. Second category is composed of phrases where second word is head and first word is modifier as in “dianhuo/ignite touchan/put into production”. Annotation of the verb phrases in this category is done for head word, again whole compound is not used for the annotation. Last category consists of multi word compounds having first word as head. Generally second word is used as particle and has little effect for the meaning of the compound. Verbs in this category are more like phrasal verbs in English and annotation is carried out for the whole compound. However, they do not have separate frames instead they are added to the head verbs frame file as a different frameset. Examples for this category is as follows, “jianshe-cheng/construct-into”, “daban-cheng/dress-as”, “kanchengshi/consider-as”, “tianjia-dao/add-to”.

Annotation in this proposition bank is done with respect to verbs and their nominalization where they share the same form. Each predicate is annotated for their core arguments (ARGNs) and also semantic adjuncts (ARGMs). Verbal predicates present syntactic variations due to the general syntactic processes like topicalization, passivization, and BA-construction in Chinese. Because of these variations, some arguments can be dislocated and end-up with dependencies between the verb and the dislocated argument. For these kind of arguments empty categories are present adjacent to the verb and annotated with the same argument tags as dislocated argument.

Also, annotation of verbal predicates, having prepositional phrase is ambiguous in Chinese. PP-attachment problem for English is defined as a prepositional phrase can be attached to the verb or noun phrase object to the verb with respect to

the semantic context. In Figure 2.15, prepositional phrase is attached to verb and noun phrase in first and second sentences respectively. Both PP and NP are located after the verb in English, but this issue can be solved by attaching PP to the corresponding level i.e. VP or NP in the parse tree with respect to semantic context. On the other hand, In Chinese, prepositional phrases always occurs before the verb where NP is after. Hence the intervening verb prevent PP and NP to compose a constituent and this problem is not solvable with syntactic structure. Therefore PP arguments are not taken into account as argument or adjunct for the verbs, but regarded as related with nominalized predicates. Same situation can occur with nominal predicates where the intervening word is light verb or semi-light verb. In this situation, PP is considered as the argument of nominal predicate.

(4) I ate a pizza with friends.
I ate a pizza with onions.

Figure 2.15: PP-attachment problem In English.

Another complex situation for verbal predicates is "discontinuous arguments" where an argument in one sentence is split into, generally, two parts but having the same meaning. Both cases should be treated similarly for the argument annotation so splitted argument is tagged with the same argument label. When the subject is splitted with predicate, both part is tagged with ARG0 and the predicate split is also tagged with -PRD which is given ARG0-PRD label. Also, discontinuous arguments are caused by subject-predicate verb compounds where the first verb is semantic subject of the second verb and the subject of the first verb is the subject of the whole compound. Likewise a possessorpossessee split can occur in the sentence where ARG1 is splitted into two different parts and annotated separately. Same problem exists in multi participant verbs where the event is done by multiple parties. These discontinuous arguments are also labeled with same argument label.

Along with the verbal predicates, nominalized verbs are annotated in the Chinese PropBank as part of the Chinese Nombank project. Although the English NomBank project annotates relation nouns as predicate, Chinese counterpart limits the nominal predicate concept to nominalization of verbs. Aforementioned the forms of the verbs and their nominalizations are in the same form in Chinese which simplifies the annotation of nominalized verbs. Same frame files can be used for both predicate types without any change. However, in some cases nominal versions of the verbs do not bear the same senses with their verb forms. The content of the nominal predicates are found by checking all the nouns which matches with a verb written in the same form. Nevertheless, some of the matching nouns do not share the same arguments with their verb forms, so they are omitted and not annotated. Also, in some cases certain senses of the nouns are not nominalizations such like Professor when it is used as title. On the other hand, when it is used as "English Professor", it is nominalization of verbal form, so counted for annotation.

There also exist limitations for the argument selection of the nominalized verbs. Some of the modifiers and arguments are not annotated since they are noun form specific arguments. Any argument does not occur in verb form is out of annotation. Determiner phrases and quantity phrases which are expressions of duration and frequency are excluded since they can be used for verb forms too. Also, nominalized verbs can be the head of a relative clause, in that case relative clause generally is not an argument of the nominalized predicate.

Prior to the manual annotation, a semantic tagger which utilizes the syntactic variations to map these informations to an initial argument label is designed. As a first step subcategorization frames are extracted for each predicate instance in the frame files. In this manner all syntactic cases are gathered which then be used for mapping to the predicate-argument structure. The functional tags in the CTB are used to determine argument types; -SBJ (subject), -OBJ(object), and -DIR (direction) are core argument candidates, and -TMP (temporal), -LOC (location), -MNR (manner), -PRP (purpose and reason), -CND (condition) are

probable adjuncts. Adjuncts are mapped directly by adding prefix "ARGM-", however core arguments are ambiguous and mapped with rules. Whenever multiple roles found for an argument, system selects an arbitrary role among found argument labels. This procedure gives 0.76 precision for all annotations. 0.87 precision for core arguments. After this step, human annotators can check and annotate any missing labels.

500,000 words from Chinese Treebank corpus is used for double-blind annotation and adjudication. 11,765 frames having 12,555 framesets are generated.

### 2.3.4 Dutch PropBank

A Dutch PropBank is constructed in [15] by using SoNaR1 corpus which includes one million written Dutch words. SoNaR1 corpus is subcorpus of SoNaR [49]. The corpus contains texts from six different genres, which includes administrative texts, autocues, texts treating external communication, instructive texts, journalistic texts and wikipedia. 500K words from the corpus first manually annotated with PropBank guidelines style for Dutch and remaining 500K is automatically annotated by using manually annotated corpus for training. An existing Dutch semantic labeler is retrained with 2,000 manually verified sentences and used to pretag semantic roles. After that, experiments are conducted for fine-tuning the performance.

For the manually annotation of the first 500K words, English PropBank frames are labeled and used instead of creating new Dutch frame files. In the annotation phase TrEd tree editor [50] is used. After the annotation of initial 50K words, annotations are manually verified and after 300K annotations are checked. In these error analyses, semantic labeler is observed to have difficulty with annotations of higher numbered arguments (ARG3-ARG5). Also, for annotators, it is hard to select true English frame files for Dutch verbs. To solve this problem the name of the English frame file is added to data.

As for the automatic annotation of remaining 500K words, different training methods are used to maximize performance. Labeler is trained with each genre individually and with a collective dataset including samples from all genres. Training on large data maximizes f-measure, on the other hand having genre specific training dataset is important for classification optimization. Small amount of in-domain training is found to be sufficient and low number of unique predicates for the particular genre yields better performance. However, this low number decrease generalization power of that particular genre. Also, including enough out of domain data is important to have a robust labeler.

### 2.3.5 Finnish PropBank

The Finnish PropBank [16] is built on Turku Dependency Treebank (TDT) [51] consisting of 15126 sentences and 204399 tokens. It contains 10 different genres such like Finnish Wikipedia, financial news and amateur fiction. Syntactic analyses of the treebank are annotated with Stanford Dependency. These analyses include two layers where first layer contains analyses in tree format, and the conjunct propagation and additional dependencies layer. In the second layer, additional dependencies are added to treebank. There are three analyses in the second layer which can support PropBank annotation. First one is "Propagation of conjunct dependencies", where head word is marked in a cordination if it is the first element. Via these information, modifiers of the head can be distinguished from modifiers of some conjuncts. Next one is "External subjects", if a word is shared as subject by multiple predicates, only one tag is annotated in the base layer. Other subject annotations are stored as external subject in the second layer. Last phenomena of the second layer is "Syntactic functions of relativizers" where secondary syntactic function of phrases having relative word is marked. Also in the treebank, morphological analyses gathered from OMorFi are included. If a word has morphological analysis as verb in the possible analyses list received from OMorFi, it is selected for annotation even it is not selected as

verb in the treebank. As a result redundant words are examined as verb but this gives higher recall.

In the beginning of the annotation double annotation methodology is used where the same sentence annotated two times and annotation the merged. This procedure is used to set rules for possible complex situations. Then after the learning phase finished, single annotation takes place for the remaining sentences to speed up. Also, verbs with high frequency are double annotated partially where rare verbs are completely annotated by a single annotator. Annotations are completed by using a custom editor. Editor have two functionality for frameset construction and annotation phase. Annotators can mark annotations as "unsure" whenever they are not sure. Also, if the predicate examined is not a valid verb, it can be tagged as "not a verb". Furthermore tool also has options for copying framesets for suitable verbs. This can be done in batch mode for multiple verbs or with copy mode. After framesets copied from an existing frameset, framesets marked as connected. This information can be used to gather extra examples from similar verbs for verbs with rare occurrence.

In the generation of Finnish PropBank, English PropBank is utilised, some of the framesets are in the same structure with the English language. However, there are also some differences between two language. Especially some Finnish verbs have different interpretation. In causative derivations, some verbs like "move" in English, can be used with an agent subject and patient, with only an agent as subject, or with a patient subject. On the other hand instead of that usage, two different verbs are used in Finnish. The corresponding Finnish verb "liikkua" (to move) is intransitive and can be used with only an agent as subject, or with a patient subject but can not be used with agent subject and patient. Instead, the verb "liikuttaa" (to make something move) is used for transitive case.

Furthermore reflexive derivations are different in both languages. These derivations can be used in two ways. Either agent performs an action to himself or the situation expressed by the verb occur automatically. For the first type transitive

verb “pukea” (to dress) can be given as an example. This verb take “-utu” affix and becomes “pukeutua” (to get dressed or to dress oneself). Same meaning can be acquired by using the root verb and oneself together. Derived verb is paired with “to dress” in English PropBank but frameset of the derived verb does not contain ARG1 since it does not take another argument for dressing action instead ARG0 is dressed by ARG0. The second reflexive derivation is for automatic events where the event happens by itself. As an example “sulkeutua” can be derived from “sulkea” (to close), which can be used with unknown agents. In this type of derivations, verbs do not have ARG0 in their framesets.

There are also contextual cases which are different in the underlying treebanks instead of differences between languages. The “juosta” which can be translated as “to run” is an example for this kind of difference. Commonly used meaning for the verb in Finnish is running from a location to another. On the other hand, there is no frameset in English PropBank for “to run” that contains this arguments.

After the construction of this proposition bank, a Finnish automated semantic role labeler is also implemented using machine learning systems.

### 2.3.6 French PropBank

A proposition bank for French is also constructed by using semantic annotations for English in [17]. Study proposes semantic annotation scheme is cross lingually valid and lexicon generated for a language can be used for another language. In order to prove this hypothesis French sentences are manually annotated using English frame files. These annotations constitutes a gold standard for the syntactic-semantic parsing task afterwards. The validity of the semantic annotation scheme simplify how the annotators achieve consensus. Thus the parallelism between languages can be measured with inter annotator agreement, distributions of disagreements, and amount of predicate labels does not transfer. Most of

the roles in the annotation scheme of English fit to the French sentence without making adjustments. Divergent cases mostly occur in idioms and collocations.

Annotators first find English translation of French predicates, whenever they can not find an appropriate word, they use dummy label. TrEd editor [50] is used to annotate sentences. 1040 sentences are selected from the parallel sentences in Europarl corpus. 1040 sentences are divided into three parts, 40 for two training phase, 100 for calibration and 900 for the main annotation task. Inter annotator agreement is calculated in each phase. Inter annotator agreement for predicate annotation is very low 59% if calculated in verb sense level. If the different verb senses are count true as if comparison is made at verb class level, agreement rises to 81%. However, using verb sense is more accurate for cross language validation. Another measure for cross validity is the percentage of dummy predicates in the annotation. 82% of the dummy predicates are French multi word expressions. Also, idioms and collocations are problematic for predicate disagreement.

### 2.3.7 German Frame Based Lexicon

A semantically annotated corpus for German Language is presented in [18], [19]. It is annotated as a part of the SALSA (SAarbrücken Lexical Semantics Annotation and analysis) project and named as SALSA corpus. The Tiger corpus [52] which contains 80,000 sentences and 1.5M words from German newspapers is used as basis for the annotation. Unlike PropBank projects, corpus is annotated in the framework of Frame Semantics. A light weight German FrameNet is constructed and frame elements specified for the verbs occuring in the corpus. Frames from English FrameNet are reused in the constructed FrameNet. Most of the time, frame structures between these languages are similar with the help of minor changes and extensions. Also, word sense disambiguation should be done prior to annotation. In the sense selection German WordNet is used. Some of the word senses are not covered by the English FrameNet yet. FrameNet does not supply a frame for these words so proto-frames are added for the missing frames.

After constructing FrameNet for the corpus and word sense disambiguation, annotation phase is started with respect to resources prepared. In the annotation process, for each sentence frame evoking elements are found and sentence is annotated with respect to frame elements. In the first release of the project, 20380 instances are annotated for verbal predicates, which is covered 493 verb lemmas. Also, there exists 348 annotated instances for 14 noun lemmas. In the second release of the project more than 17,000 instances are annotated for nominal lemmas.

### 2.3.8 Hindi PropBank

Vaidya *et al.* are presented a Hindi PropBank (HPB) [20]. It is annotated on top of Hindi Dependency Treebank (HDT) [53] with predicate argument structure. The dependency structure is suitable for the analysis of flexible word order of Hindi language. Corpus consists of 32,300 words and 160 frameset files are generated. Frame files are not created separately for light verb constuctions. Annotation of the dependency treebank is done by using Cornerstone and Jubilee [41], [42] and semantic arguments at the chunk level are annotated using verbs syntactic dependents. Some of the arguments like ARG2 is subdivided into labels such as ARG2-ATR (attribute), ARG2-GOL (goal), ARG2-LOC (location) and ARG2-SOU (source) to prevent ARG2 labels to become semantically overloaded. Empty arguments are also annotated in HPB, "PRO" (empty subject of a non-finite clause), "RELPRO" (empty relative pronoun), "pro" (pro-drop argument), and "gap-pro" (gapped argument) tags are used for the corresponding cases.

Also, they propose a probabilistic rule-based system for tagging arguments using syntactic dependents. The rule-based system classifies 47% of PropBank arguments with simple rules. Prior to explaining rules, a mapping from dependency treebank annotation to proposition bank annotation is shown in Table 2.5.

Although a mapping is provided for argument annotation, some cases are differently interpreted in HPB and HDT. The example sentences in Figure 2.16 show

Table 2.5: Argument mapping between Hindi Dependency Treebank (HDT) and Hindi PropBank (HPB).

| HDT Label | HPB Label |
|---|---|
| k1 (karta); k4a (experiencer) | Arg0 |
| k2 (karma) | Arg1 |
| k4 (beneficiary) | Arg2 |
| k1s (attribute) | Arg2-ATR |
| k5 (source) | Arg2-SOU |
| k2p (goal) | Arg2-GOL |
| k3 (instrument) | Arg3 |
| mk1 (causer) | ArgC |
| pk1 (secondary causer) | ArgA |

the notion of unaccusativity of verb "to break". *The boy* and *The window* are tagged as k1 in HDT, however *The boy* is ARG0 and *The window* is ARG1 for PropBank annotation. Mapping from k1 to ARG0 is complex for these kind of verbs.

(5) The boy *broke* the window.
The window *broke*.

Figure 2.16: Example sentences for unaccusativity.

Furthermore, there is a similarity between modifiers in PropBank and definitions of some HDT labels. Since modifiers are not verb specific, high mapping accuracy is expected. In the Table 2.6 the mapping between modifiers and HDT labels are presented.

Syntactic dependents are used to identify arguments of predicates. Via this heuristic,argument identification has a precision of 99.11 %, recall of 95.50%, and an F1-score of 97.27%. Since HDT and HPB are annotated with the same principles, these high rates are expected. Only 4.5% of arguments are not found by the system which are mainly empty arguments in PropBank annotation. For the argument classification, different types of rules are used. A deterministic rule for classification of ARGM-PRX arguments is used to classify words which has "pof" dependency with the predicate. ARGM-PRX's are argument-predicating

Table 2.6: Modifier mapping between Hindi Dependency Treebank (HDT) and Hindi PropBank (HPB).

| HDT Label | HPB Label |
|---|---|
| sent-adv (epistemic adv) | ArgM-ADV |
| rh (cause/reason) | ArgM-CAU |
| rd (direction) | ArgM-DIR |
| rad (discourse) | ArgM-DIS |
| k7p (location) | ArgM-LOC |
| adv (manner adv) | ArgM-MNR |
| rt (purpose) | ArgM-PRP |
| k7t (time) | ArgM-TMP |

expression which is used in complex predicates. Furthermore, empirically-derived rules are generated with respect to the statistics of dependency features. Three features are used for the generation of the empirical rules, predicate ID, predicates voice type, and arguments dependency label. These tuples are used to decide the probability of the PropBank annotation. Probability is calculated by estimating a maximum likelihood of each PropBank label being associated with the feature tuples. Also, low probabilities are filtered with a threshold calculated by a ten fold cross-validation. Finally, linguistically motivated rules are created with respect to linguistic intuitions. The correlations between syntactic and semantic arguments are captured and rules generated for these correlations manually. These rules for numbered arguments then stored in the frameset files. Most of the rules are already found by the empiricaly derived rules but rules for the verbs that do not occur in train set will not be generated by the previous method. The evaluation for the rules are as follows, deterministic rules yields 94.46% precision and 100% recall. For the precision 5.5% of the complex verbs do not match with the "pof" relation in HDT. Empirically derived rules are also yields 90.37% for precision, 44.52% for recall, and 59.65% for F1-score with a threshold of 0.93. And lastly, linguistically motivated rules gives better result for ARGN annotations which improves total recall to 55.28% but decrease precision to 89.90% and F1-score rises to 68.44%.

### 2.3.9 Japanese Relevance Tagged Corpus

Another example for semantically annotated corpus is Japanese Relevance Tagged Corpus [21]. Kyoto University corpus is annotated for relevance tags such as predicate-argument relations, relations between nouns, and coreferences. This corpus contains 40,000 syntactically tagged sentences. An annotation tool which was constructed in Kyoto University corpus project is used for tagging sentences. PropBank style arguments are not used but since predicate-argument releations are tagged, corpus can be regarded as a proposition bank. Also, argument annotations can be converted to PropBank style easily.

In the first phase, two annotators tagged 1,000 sentence to establish a specification for annotation. After the specification established, constant annotation is started. Tags obtained automatically from their case and ellipsis analyzer are modified by the annotators. Annotations is done for three types of relevance. As for predicate-argument relations, predicates are tagged with an argument word and the relation provided by case markers. In Japanese, postpositions are served as case markers. Examples of postpositions of the language is "ga" (nominative), "wo" (accusative), and "ni" (dative). In Figure 2.17, an example annotation for predicate-argument relation is presented. Predicate "yonda" (read) is tagged with "Taro" and "shimbun" (newspaper) as ARG0 and ARG1 in PropBank.

(6) Taro - ga shimbun - wo yonda.
Taro nom newspaper acc read
(Taro read a newspaper.)
yonda ⇐= ga :Taro, wo:shimbun

Figure 2.17: Japanese sentence is annotated for predicate-argument relation.

Surface cases are used for relations instead of deep cases, since boundaries of the relations are not clear. Also, using deep cases results in difficulty to form set of relations and select one relation for annotation. As stated previously, tags are provided automatically by analyzers prior to annotation and annotators modify any erroneous tags. Most encountered incorrect automatic annotations are due

to two characteristics of Japanese: disappearance of case markers and omission of arguments (zero-pronouns). In these cases automatic analyzers are failed to find correct tags so the annotators should correct these erroneous annotations. Moreover predicate-argument relations are extracted for the nouns bearing action meaning like PropBank. Noun "nyuugaku" (admission) is annotated in the sentence in Figure 2.18 since it means action of "admitting", so it is tagged like verb "admit".

(7) Kare-no daigaku nyuugaku - wa yoi news da.
his university admission TM good be
(His admission to the university is good news.)
nyuugaku ⟸ ga:Kare, ni :daigaku

Figure 2.18: Japanese sentence is annotated with respect to noun bearing action meaning.

In the annotation phase, annotators should also consider some problematic cases for Japanese. Japanese sentences do not have word segmentation. Some compound words contains multiple words like "hounichi" which means "hou" (visit) and "nichi" (Japan). For such cases word segmented into pieces to simplify the annotations, so annotator should also consider the correctness of the segmentation. The next problematic case is tags with multiple arguments are distinguished with respect to meaning. If the arguments are obligatory, and all of them is coordinate, they are tagged with "and" relation. "And" relation is also used for arguments meaning alternative to each other. If arguments are not coordinate and their predicate is attached with a case like "with", they are tagged with different case markers. Another case for multiple arguments is if any of the words in the tag is proper, they are tagged with "or" relation to indicate that one of the elements is a referent. Another complex case is noun modifying clauses where a predicate-argument relation exists but no case marker is found in the sentence. Annotators should tag these nouns for predicates even in the absence of postpositions. Lastly if an argument is not a specific entity, but people without antecedents expressed should be tagged with "Unspecified people" tag.

The corpus is also annotated for noun relations and coreferences and without its style it contains similar semantic information like the other proposition banks. Currently 1300 sentences are annotated.

### 2.3.10 Korean PropBank

There are two different studies in terms of Korean PropBank. Palmer *et al.* built Penn Korean PropBank [22] using Korean English Treebank Annotations and Korean Treebank Version 2.0. Annotation is done with respect to verbs and adjectives appearing in the Treebank. The words near these semantic predicates are annotated with argument and adjunct roles. Penn Korean PropBank consists of 2749 frame files and two annotation files. In Korean English Treebank, 9588 predicates occurred in over 54,000 words are annotated. In the second file, 131,000 words from Korean Treebank Version 2.0 containing 23,707 predicates are annotated. Also, Song *et al.* construct a proposition bank [23] on top of syntactically annotated corpus of Korea Electronics and Telecommunication Research Institute. In this study more than 3,000 sentences are annotated by using frame files generated in Penn Korean PropBank [22].

### 2.3.11 Multilevel Annotated Corpora for Catalan and Spanish

A preliminary multilevel annotation for the Catalan and Spanish is completed in SemEval-2007 and reported in [26]. Later AnCora, a multilevel annotated corpora for these languages is presented in [27]. AnCora is built with the sentences from newspaper and newswire articles previously collected in 3LB [54] and CESS-ECE corpora [55]. AnCora is divided into two sub-corpora, AnCora-Es and AnCora-Ca where Es represents Spanish and Ca is abbreviation of Catalan languages. Both corpora contains 500,000 word tokens annotated for morphological (PoS and lemmas), syntactic (constituents and functions), and semantic (argument structures, thematic roles, semantic verb classes, named entities, and WordNet nominal senses)in different levels. Annotation process for each level is different

and fulfilled in automatic, semi-automatic and manual fashion. The annotation of different levels is completed in order from the lower to upper layers.

In terms of proposition bank perspective, semantic level annotations especially argument structures are annotated. Annotation is carried out in semi-automatic fashion. First syntactic functions are mapped to the semantic arguments automatically, then annotators check the automatic annotations and tag any ambiguous unannotated word. For the argument structure of verbal predicates, numbered arguments from ARG0-ARG5, ARGM's, ARGA and ARGL is used where ARGM's are used for adjuncts, ARGA represents external agents and ARGL is used for the complement word of light verb constructions. 187,000 word tokens from AnCora-Es are completely annotated, and all the words for Catalan language (AnCora-Ca) is annotated. Along with the annotation in PropBank style, thematic roles are also annotated for the corpora in separate layer.

### 2.3.12 Persian PropBank

Mirzaei and Moloodi presents manually annotated Persian PropBank, PerPB [24]. It is constructed on Persian Dependency TreeBank (PerDT) [56] which consists of 29982 syntactically annotated sentences. Head information for each word and sentence is included in the PerDT. From the head and dependent information, words to be annotated are selected automatically. Nominal and adjectival predicates are also tagged alongside the verbal predicates. Nominal and adjectival words should carry the following specifications in order to be annotated for predicate-argument relation. If a propositional noun or adjective have at least one visible argument or one visible adjunct in the sentence, it is then taken into consideration for annotation.

There exist differences between PerPB and other proposition banks. In opposition to the other PropBanks frequency adverbs like sometimes, always, never and repetition word again is tagged with the functional tag REPETITION in PerPB

whereas these verbs are used mostly in TEMPORAL function tag in other PropBanks. Also, negation is not a separate constituent in Persian instead it is a prefix (na...) inserted before or after the predicate as in Turkish. Thus the negation should be examined with morphological analysis prior to annotation. In other proposition banks, conditionals are tagged with ADVERBIAL function tag where it is tagged with CONDITIONAL label in PerPB. Modal verbs in PropBank are tagged with MOD label, however in PerPB, any formation of the epistemic or evidential concepts like perhaps, may, its possible, etc. are tagged with MOD. Furthermore, syntactic movements to express some discourse functions may occur in Persian language since it has free word order. There are two labels for these movements, TOPICALITY and RELATIVE CLAUSE LINK. Topicalized words are annotated with the former, and latter is used to label relative clauses. If the relative clause is adjacent to its head, RCL0 is used, otherwise if there exists other constituents in between, it is tagged as RCL1. Also, prior to the annotation procedure, no frame files are created in PerPB. PropBank and VerbNet approaches combined for semantic role annotation. Words are annotated with PropBank labels and also with respect to semantic role approach of VerbNet. Instead of frame files, Persian Semantic Valency Lexicon is constructed simultaneously as the corpus is tagged.

Some of the sentences are double annotated, if there exist any differences between the annotations, they are reported to the two supervisors. Then the correct annotation is selected or a new annotation is tagged. The double annotated sentences are then used for the calculation of inter-annotator agreement.In the study 29,982 sentences (all of the sentences provided in PerDT) are annotated where total number of distinct verbs is 9,200. Also, 1,300 nominal and 300 adjectival predicates are annotated.

### 2.3.13 Portuguese PropBank

PropBank for Portuguese language is constructed in two different study. Duran and Aluísio proposed Propbank-Br [25] by annotating semantic information on a Brazilian Portuguese Treebank. The corpus consists of 4213 Brazilian Portuguese sentences from a subcorpus of Floresta Sintá(c)tica [57]. Frame file generation prior to the annotation is not carried out for this study. Instead, frame files from English PropBank is used to tag Portuguese sentences. Since no frame file is generated in advance, instead of Cornerstone and Jubilee [41], [42], SALTO [58] an annotation tool created for German FrameNet project is adapted and used. Only the last verb in verb chains is used for annotation task, the auxiliary parts are not taken into account. Some sentences in the corpus is repeated with respect to the verb count they contain to create an instance for annotation of each found verb. After that, the number of sentences increased to 6142 from 4213 sentences.

In the annotation, verbs found in Portuguese sentence is searched in English frames and sentence is annotated with the roles provided in the corresponding English frameset. This procedures have some drawbacks since some verbs active multiple English frames. Also, matching frames may not contain the necessary arguments. As an example the Portuguese verb "renunciar" is synonym with *relinquish*, but a common beneficiary role for Portuguese is not included in the frameset of English word. For these missing roles a suitable argument role is assigned for the annotation. Furthermore, some phrases like "no meu aniversário" (in my birthday) are ambiguous for argument selection. Here the argument can be tagged as temporal adjunct considering meaning date of the birth, or as locative with meaning birthday party. These ambiguous cases should be clarified in the guidelines otherwise annotators may select different arguments which reduces inter annotator agreement. On the other hand some problems rises because of Portuguese parser output. Parser used in this study does not produce traces like The Penn Treebank has for suppressed syntactic elements. These traces are also used for the annotation in English PropBank. For PropBank-Br whenever

an empty category is found in an embedded clause, a role label is assigned to the proper constituent, although it is in the main clause. And these clauses are also flagged as ELIPSE for machine learning approaches. Likewise, correfering constituents is tagged with respect to the referred constituent since corpus output does not have correference resolution. So pronouns correferring an ARG0 constituent or adverbials which have correference with an ARG1 constituent is tagged with the same argument. Beyond these problems, sometimes syntactic boundaries do not match with the argument annotations, that is a syntactic constituent may contains multiple arguments or an argument may be spanned over multiple syntactic constituents. In the first case syntactic constituent is tagged with the role label with higher precedence where the precedence is from ARG0 to ARG5 and ARGNs to ARGMs. As for the second multi syntactic arguments, all the syntactic constituents are tagged with the same label. In this study 6142 instances are annotated for 1068 predicates. In the next step, creation of frame files for these predicates are planned. Also, they trained a semantic tagger by using these annotated corpus.

Branco *et al.* also generated another proposition bank for Portuguese, The CINTIL-PropBank [59]. It is constructed by using CINTIL-DeepGramBank and contains about 10,000 sentences. Initially proposition bank is constructed semi-automatically by using deep linguistic grammar. LXGram, a grammar for the computational processing of Portuguese is used to obtain parses. These parses then combined with the parses manually selected by human annotators and CINTIL-DeepGramBank is composed. Two annotator then select the true analysis among these deep grammatical representations, whenever their decisions contradicts, a third annotator determines the true analysis. Afterwards, syntactic constituency trees are obtained by processing the selected true analysis from the CINTILDeepGramBank. The tool lkb2standard [60] is used for the extraction of syntactic trees from the files exported by [incr tsdb()] [61]. These extracted trees are then used to construct CINTIL-PropBank. Some of the semantic roles (ARGNs) are generated directly from the deep grammatical representations. The

remaining adjunct roles are then annotated manually. Tools are developed to convert trees to excel sheets for facilitating adjunct annotation and reverse excel sheets to trees. Again two annotator select the correct annotations and third annotator is only involved in whenever annotations differs from each other. At the time paper published CINTIL-DeepGramBank contains 5422 sentence and 4047 remaining sentences are being processed, so only the 5422 sentence used for annotation.

### 2.3.14 Urdu PropBank

The first study proposes a PropBank style labeled corpus for Urdu is [32] where SRL annotations are generated using English-Urdu parallel corpora. This parallel corpora contains 6000 sentences translated from 317 WSJ articles of Penn Tree Bank. 2350 sentences among this set has PropBank annotation and processed in this study.

Urdu, an Indic language, is grammatically different and shares very little vocabulary with English. However, as stated in frame semantics paradigm, semantic frames are constructed on conceptual structures so they have less tendency for syntactic variations. Study proposes the semantic correspondence between Urdu and English enables annotation transfer applicable. The projection procedure is as follows, for each sentence in the parallel corpus, they first annotated English sentence with semantic roles and then transfer annotations to the Urdu language using lexical information, word alignment and linguistic rules about syntax of the word. During this transfer, challenging situations occurs such like thematic divergence where a theme in subject position in the English sentence changes to the object in Urdu. An example is provided in Figure 2.19. Conflatational divergence, translation of an English word is a group of words, “plays” is translated to “kirdar ada” in Urdu. Demotional divergence and structural divergence occurs when sentences are dissimilar syntactically and results as random matchings as in Figure 2.20. Prior to word alignment, a sentence alignment task is completed.

To define sentence boundaries, a two step approach is used. In the first step word counts in each sentence is calculated by using the word occurences of important PoS categories. PoS categories like proper nouns, adjectives and verbs are taken into the account to define sentence boundaries. This method also resolves conflatational divergences where translation contains multiple words. Next Urdu-English lexicon is used to obtain English words from Urdu sentences. In order to accept alignment of the sentences, English-Urdu sentence pair with maximum lexical similarity is selected. Word alignment is then achieved by using both translations English-Urdu and Urdu-English. Berkeley Aligner package which is used to choose bidirectional alignments. Word alignment ratio is calculated as 71.3% over 200 sentences annotated manually. However, not all of the words aligned and some words in English can be translated as multiple tokens in Urdu like verbs so relying on only the word alignments result in having erroneous projections. To overcome this situation POS, tense and chunk information are used to project annotations. As a first step in the annotation projection, predicates are transferred. Then after identifying predicate roles argument annotations are transferred. Here the success of predicate identification determines the annotation projection for arguments. Chunk boundaries tagged in Urdu words used to improve predicate recognition. English sentence do not have chunk boundaries instead PoS tags like IN, TO, MD, POS, CC, DT, SYM is used to separate virtual boundaries. The first rule for predicate recognition is Urdu-English translation, if any word is found in the dictionary, then the annotation transferred to the verb chunk. Also, approximate English chunks is used to align Urdu verb chunk for annotation projection. Also, tense information is encoded in the PoS tags of the verbs in both English and Urdu. This similarity is exploited to match predicates. After transferring predicate annotations, annotation projection for arguments starts. Word alignments is not necessary for proper nouns. Matched proper nouns are annotated, then rest of the arguments are determined with respect to word alignment and PoS tags.

Annotation projection is compared to 200 sentences manually annotated. 100 of

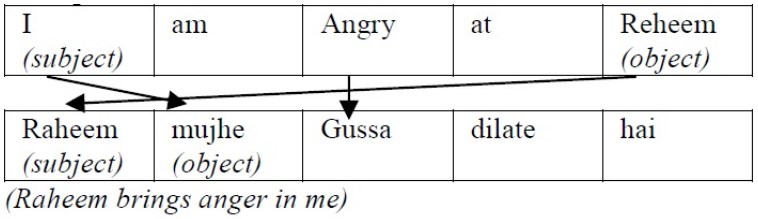


Figure 2.19: Example sentences for thematic divergence.

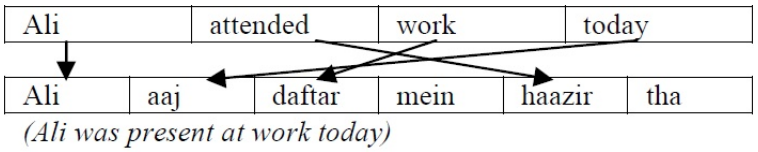


Figure 2.20: Example sentences for demotional and structural divergence.

them are long sentences which contain multiple predicates and having more than 55 words. The remaining 100 sentences are short sentences having about 40 words at most and no complex predicate. Precision for long sentences is 77.8% and 92% for short ones for all tags. Precision for predicates is 68% and 80.9% respectively for long and short sentences. Also, verb predicate and argument detection models are trained using automatically annotated Urdu corpus.

There is also another study [62] which constructs a proposition bank for Urdu on top of Urdu Dependency Treebank [63]. In this study, frame files is not generated from scratch, instead frame files suitable for Urdu is ported from Hindi and Arabic PropBanks. Urdu and Hindi languages are in the same family and very similar languages. The differences in structure and vocabulary between these languages are mainly because of most of the vocabulary used in Hindi is derived from Sanskrit and conversely Urdu gathered words from Arabic and Persian. Predicates shared by these languages are examined in detail in [64]. These shared predicates are ported and used for Urdu proposition bank.

Annotation phase is then studied using the annotation tool Jubilee [42]. As in the Hindi PropBank [20] ARG2 tags differentiated to sub labels to prevent ARG2 label to be overloaded semantically. Moreover morphological causative arguments

are tagged with ARG-A tag. Complex predicates where nominal elements complement the verb is tagged with ARGM-PRX. Annotation procedure faces with some difficulties such like ARG0 and ARG1 distinction. Since Annotation tool provides information about dependency trees, annotators affected by the labels of dependency relations. To solve this issue, arguments having agentivity is tagged with "ARG0" and "ARG1" is used or the arguments clearly affected by the action. Unergative and intransitive verbs are annotated with "ARG0". Animate subjects that has voluntary control on the action is ARG0 for intransitive verbs (like *nAca* "dance" and *xORa* "run"). Also, the verbs *hai* "be" and ho "become" do not have ARG0 in their frame files. However, annotators tend to tag arguments lacks agentivity or not involved in volitional action as ARG0. Some words in the language can be tagged as differently according to context they are used in which further complicates the annotation. 44,000 words corpus is double annotated and annotations compared each other. The Fleiss' kappa is calculated as 0,88 which is a high degree in agreement. Annotated corpus contains 180,000 double annotated tokens by two annotators, and another 100,000 tokens annotated for complex predicates.

### 2.3.15 Summary

We presented proposition bank studies for various languages in this chapter. A brief summary can be seen in Table 2.7, for each language we tried to give information regarding the size, year of the research, used annotation tools and methods for annotation and frame creation in a nutshell. We also explained English PropBank in detail in Chapter 2.2 and we added English to the summary table. The latest release of the English proposition bank dataset contains nearly 1,5 million words. 7312 unified frame files exist in the github page of the project. Project started in 2002 and the last release date for the annotated corpus is 2013.

Table 2.7: Proposition Bank studies for different languages.

| Language | Size | Year | Tools Used | Method |
|---|---|---|---|---|
| Arabic | 1955 frames, 2446 framesets | 2008-2010 | Jubilee and Cornestone | Manual Annotation |
| Basque | 37,000 words for 32 verbs | 2006-2010 | AbarHitz | English Frames used<br>Semi-Automatic Annotation |
| Chinese | 500,000 words, 11,765 frames, 12,555 framesets for Verbs and their nominalization | 2006-2009 | - | Semi-Automatic Annotation |
| Dutch | 500,000 words manual 500,000 words automatic | 2012 | TrEd | English Frames used<br>Semi-Automatic, Automatic Annotation |
| English | Nearly 1,5 million words, 7312 unified frames | 2002-2013 | Jubilee and Cornestone | Semi-Automatic Annotation |
| Finnish | 15,126 sentences 204399 tokens | 2015 | Custom Editor | English Frames used<br>Manual Annotation |
| French | 1040 sentences | 2010 | TrEd | English Frames used, Manual Annotation |
| German | 20380 verb, 17000 noun instance annotated | 2003-2006 | SALTO | English FrameNet Files used<br>Manual Annotation |
| Hindi | 32300 words 160 frames | 2011 | Jubilee and Cornestone | Manual Annotation<br>Probabilistic rule-based system |
| Japanese | 1300 sentences | 2002 | Custom Editor | Semi-Automatic Annotation |
| Korean [22] | 2749 frames 185000 words | 2006 | - | Manual Annotation |
| Korean [23] | 3,000 sentences | 2012 | - | Manual Annotation |
| Catalan | 500,000 words | 2007-2008 | - | Semi-Automatic |
| Spanish | 187,000 words | 2007-2008 | - | Semi-Automatic |
| Persian | 29,982 sentences annotated for 9,200 verbs, 1,300 nominals, 300 adjectival predicates | 2016 | Custom Editor | PropBank - VerbNet Frames used, Valency Lexicon is constructed, Semi-Automatic |
| Portuguese [25] | 6142 instances for 1068 predicates | 2012 | SALTO | English Frames used<br>Manual Annotation |
| Portuguese [59] | 5422 sentences annotated | 2012 | custom editor | Semi-Automatic |
| Urdu | 2350 sentences | 2010 | - | Transfer from English Parallel Sentences |
| Urdu | 280,000 word tokens | 2016 | Jubilee | Hindi and Arabic Frame Files used |

## 2.4 Automatic PropBank Generation Studies

PropBanks are also generated automatically for resource-scarce languages by using parallel corpus. In this section, proposition bank studies for automatic generation are presented. In [65], Zhuang and Zong proposed performing SRL on parallel corpus of different languages and merging the result via a joint inference model can improve SRL results for both input languages. In the study English and Chinese parallel corpus used. First each predicate processed by monolingual SRL systems separately for producing argument candidates. After the candidates formed, Joint Inference model selects the candidate that is reasonable to the both languages. In Figure 2.21 overview of the approach is presented visually. Also, a log-linear model is formulated to evaluate the consistency. This approach increased F1 scores 1.52 and 1.74 respectively for Chinese and English.

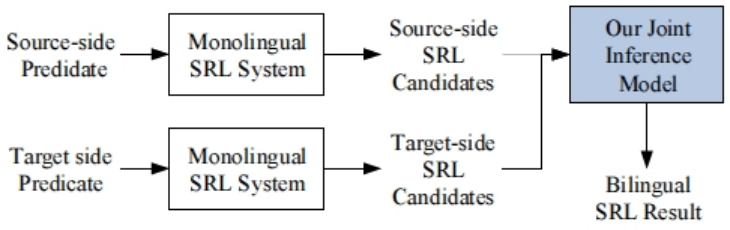


Figure 2.21: Overview of the approach [65], individually processed SRL candidates merged in Joint Inference Model and Bilingual SRL result gathered.

Van der Plas *et al.* presents cross-lingual semantic transfer [66] from English to French (see Figure 2.22). English syntactic-semantic annotations transferred using word alignments to French language. French semantic annotations gathered from the first step then trained with French joint syntactic-semantic parser along with the French syntactic annotations trained separately. Joint syntactic-semantic parser is used for learning the relation between semantic and syntactic structure of the target language and reduces the errors arising from the first step.

This approach reaches 4% lower than the upper bound for predicates and 9% for arguments.

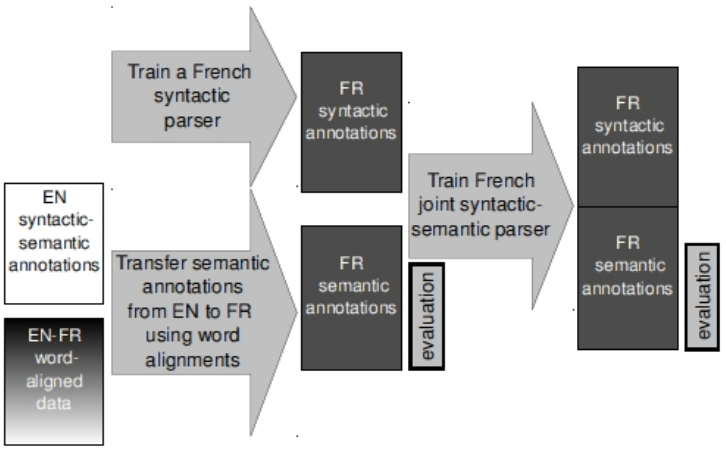


Figure 2.22: System Overview of [66], English syntactic-semantic annotation is transferred to French and result is trained with French joint syntactic-semantic parser to reduce automatic cross-lingual semantic transfer errors.

Kozhevnikov shows SRL model transfer [67] from one language to another can be achieved by using shared feature representation. Shared feature representation for language pairs is constructed based on syntactic and lexical information. Afterwards, a semantic role labeling model is trained for source language and then used for the target language. As a result SRL model of the target language is generated. Process only requires a source language model and parallel data to construct target SRL model. Approach is applied for English, French, Czech and Chinese languages.

In the next study [68], Van der Plas improves the labeling results with respect to the previous work [66] by building separate models for arguments and predicates. Also, problems of transferring semantic annotations using parallel corpus is examined in the paper. Token-to-token basis annotation transfer, translation shifts, and alignment errors in the previous work is replaced with a global approach that aggregates information at corpus level. Instead of using English semantic annotations of roles and predicate together with French PoS tags to generate French

semantic annotations, English annotations of predicates and roles used separately to generate one predicate and one role semantic annotations separately.

Gormley examines joint and pipelined methods for semantic annotation where syntactic information is not exists in [69]. Performance changes without parsing treebanks is discussed.

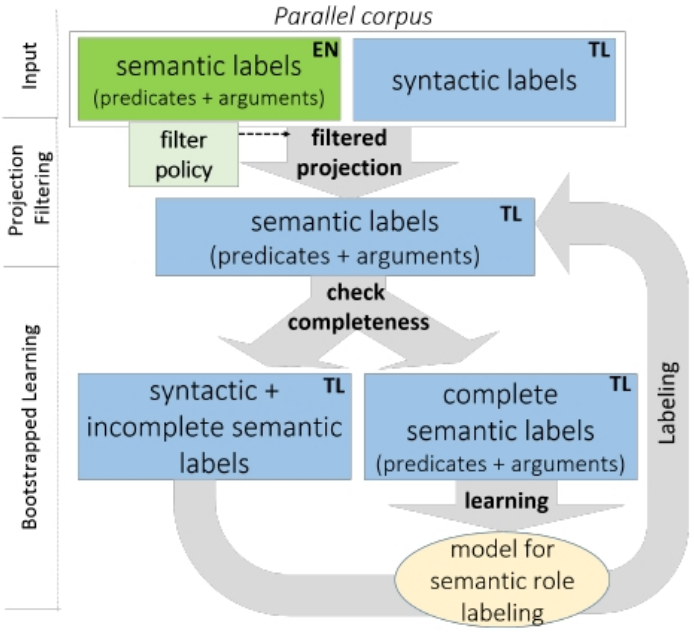


Figure 2.23: Overview of two stage approach [70]. In the first stage, conversely to the previous studies, filtered projection of high-confidence semantic labels to the target language is made. Then via bootstrap learning SRL is improved iteratively.

Akbik *et al.* propose a two stage approach [70] (see figure 2.23). In the first stage only filtered semantic annotation is projected. Since high confidence semantic labels projected, resulting target semantic labels will be high in precision and low in recall. In the next stage, completed target language sentences sampled and a classifier is trained to add new labels to boost recall and preserve precision. Proposed system is applied on 7 different languages from 3 different language family. These languages are Chinese, Arabic, French, German, Hindi, Russian, and Spanish.

# Chapter 3

# Constructed Turkish PropBank

## 3.1 Turkish TreeBank

Semantically annotated sentences are required before introducing the Propbank. In each sentence, action is expressed by a main predicate. These predicates are then used for creating frame files in the PropBank. To extract predicate-argument information, we selected 9560 sentences containing a maximum of 15 tokens from the translated Penn Treebank II [71], [72]. These include 8660 sentences from the training set of the Penn Treebank [34], 360 sentences from its development set and 540 sentences from its test set.

In Penn Treebank structure, all linguistic label stages are covered with simple format. It offers advantages like building fully tagged data set in accordance with syntactic labels, morphological labels and parallel sentences in English and Turkish. This simple adaptive format ensures automatic semantic labeling is done along with other corresponding linguistic labels.

### 3.1.1 Semantic Annotation Tool

Semantic annotation of the verbs is crucial before creating the frame files since selecting the sentences for different verb senses depends on this manual task. Most of the time, this semantic annotation task is erroneous due to the nature

of task. Having multiple annotators and controversies on the different instances of the same verb sense further increase this error rate. Any minor mistake of the annotators results in a major impact on data especially when working on a big dataset. To reduce erroneous annotations we use a in-house semantic annotation tool which visualizes sentences with their possible meanings in below of each word (see Figure 3.1). While our tool shows possible meanings for all words of the sentence, we only consider the predicate of the sentence for the PropBank point of view. So right before analyzing sentences morphologically and semantically, we detect predicates in the sentences. And afterwards, with the help of the tool, annotators easily manipulate the visualized relations.

English - Turkish sentence pairs of the verb in question are displayed in the bottom of the tool. Tag status of the Turkish sentence, whether all the words tagged or not, is represented in the colour of the sentence. Also, semantically untagged words can be automatically annotated by applying maximum likelihood with respect to the previous annotations in the corpus. These functionalities further simplifies annotators' manual annotation task.

### 3.1.2 Using Morphological Analysis for Candidate Meaning List Construction

The translated version of Penn Treebank II also contains morphologically analyzed and disambiguated words of the translated Turkish sentences [71], [72] along with the corresponding English words. In this subsection, we will explain how we used this information for the construction of possible semantic labels for a specific word.

For choosing the possible meaning of the target word, we need to extract the word form as it occurs in Turkish dictionary. Considering the morphological properties of the Turkish language, it is a big problem to decide which suffixes are used to construct word form in the sentence. A Turkish word may take many suffixes resulting in many surface forms so each word in the tree may not

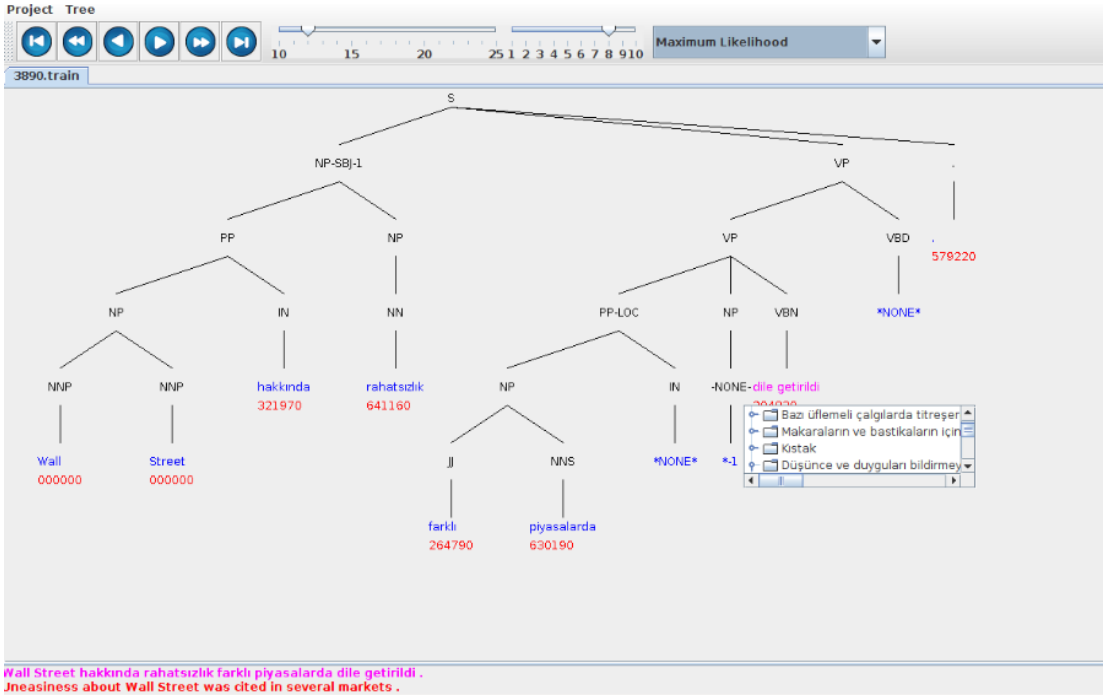


Figure 3.1: A screenshot of the semantic labeling tool : Sentence is presented as parse tree in the middle where each leaf can be annotated with the semantic meaning of the word. Turkish and English version of the sentence is also written in the bottom.

be in the Turkish dictionary. As a solution to this problem we take root form of the word with using the correct morphological analysis. Then all possible candidates are generated by appending metamorphemes. Our tool automatically analyzes words morphologically by using an analyzer implementation and provides possible morphological analyses for each word. If it finds single analysis, then automatically selects this analysis for the word. If there exists multiple analysis, annotators can select the appropriate analysis among the analyses proposed in the graphical user interface. Figure 3.2 shows an example for morphological analysis, here the root of the word is selected as the verb "düş" by the annotator among the morphological analyses.

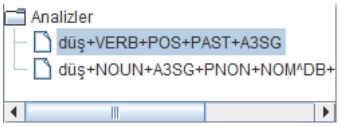


Figure 3.2: Morphological analysis of the word "düştü".

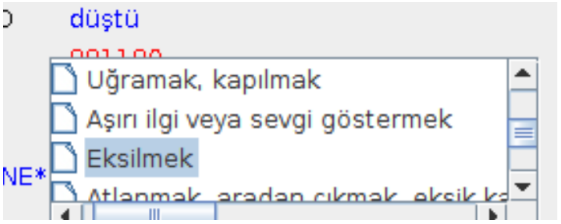


Figure 3.3: Candidate meaning list for surface form “düştü”.

After finding the correct morphological analysis, annotators use semantic editor to choose the correct verb sense. Meanings related to the root word are listed in the candidate meaning list with respect to the type of the word. In Figure 3.3 candidate meaning list of the verb “düştü” is shown since the decomposition which includes verb tag is chosen in the previous screen.

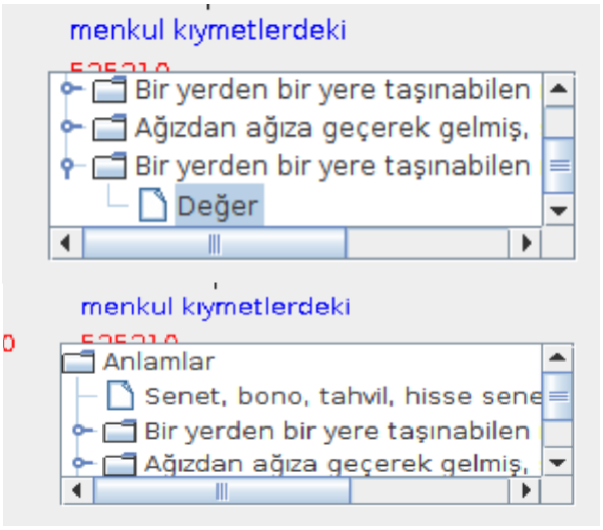


Figure 3.4: Candidate meaning list for surface form “menkul kıymetlerdeki”.

### 3.1.3 Handling Multi Words

Another problem is translation of an English word is a multi-word in the Turkish side. As explained in Section 3.1.2, finding possible meanings of individual words and then combining those meanings may not represent the meaning of the multi-word. As a solution we take both words together and apply the same approach in Section 3.1.2 and extract all possible combinations and represent them in top of candidates as shown in Figure 3.4. The top list contains possible meanings

of “menkul” and “kıymetlerdeki”, whereas first item in the bottom list shows possible meaning of “menkul kıymetlerdeki”.

## 3.2 Turkish PropBank

Semantically annotated dataset described in Section 3.1 is processed with respect to predicate-argument structure. Prior to semantic role labeling, preliminary steps is referred in the next subsection and annotation process is clarified in the latter.

### 3.2.1 Preliminary Steps

After morphologically and semantically analyzing predicates, rolesets for each verb sense in the corpora should be specified in advance for annotation process. In order to do this, a verb-sense list for semantically analysed corpora is consolidated as in Table 3.1.

Table 3.1: Sample verb sense list : Number of occurences is presented for each sense of the predicate.

| Verb | Sense ID | # Occurrence |
|---|---|---|
| işlemek | 1 | 13 |
| işlemek | 5 | 2 |
| işlemek | 8 | 9 |
| işlemek | 12 | 1 |
| işlemek | 13 | 2 |
| itmek | 1 | 1 |
| itmek | 3 | 1 |
| itmek | 4 | 2 |
| izlemek | 1 | 5 |
| izlemek | 2 | 1 |
| izlemek | 3 | 3 |
| izlemek | 4 | 6 |
| izlemek | 5 | 2 |
| izlemek | 6 | 2 |

In the next step, all sentences that contain the verb-sense instance are examined in terms of predicate argument structure and semantic roles for that sense are identified. Unlike the original PropBank frame files where each verb has a file with different rolesets for each different sense, we decide to use one xml file which contains all verbs and their senses respectively. For each verb sense, "FRAMESET" tag with the unique sense id as attribute is inserted inside the "FRAME" tags in the file. We use "ARG" tags for each role that a predicate has. "ARG" tags take name attribute as the name of the argument and semantic meaning inside. Figure 3.5 shows a part of the frame file. Each sense has a unique id and have different number of arguments with different names with respect to the semantic information.

We used seven framers in NLP group to construct rolesets. Video guidelines for creating frame files and annotation are prepared based on the original PropBank framing and annotation guidelines. These framers are educated before starting to build frame sets. Corpus is divided into seven parts in terms of verbs occurrences and each interval is assigned to a team member.

In the framing phase, the team finds out several errors such as mistranslation, false morphological analyses, and erroneous sense selections. We keep track of these errors and raise the problem to the team immediately. Translation errors are sent to the firm to retranslate the erroneous sentences. Analysis errors are solved inside the team. If the sense of the predicate is false, correct sense is selected and sentence is taken into account for extracting new sense's semantic roles.

In the frame file we create framesets for 1330 verbs, 1914 verb senses. In Table 3.2, number of occurences of the arguments and modifiers are listed. As we expected, more than 78% of the verb senses has Agent and 74% has Theme/Patient. Note that we did not cover the hidden subject situation while deciding the roles of the predicate. In Turkish, we can detect hidden subject from the inflected forms of the verbs which may increase the number of Agent role among the verb senses.

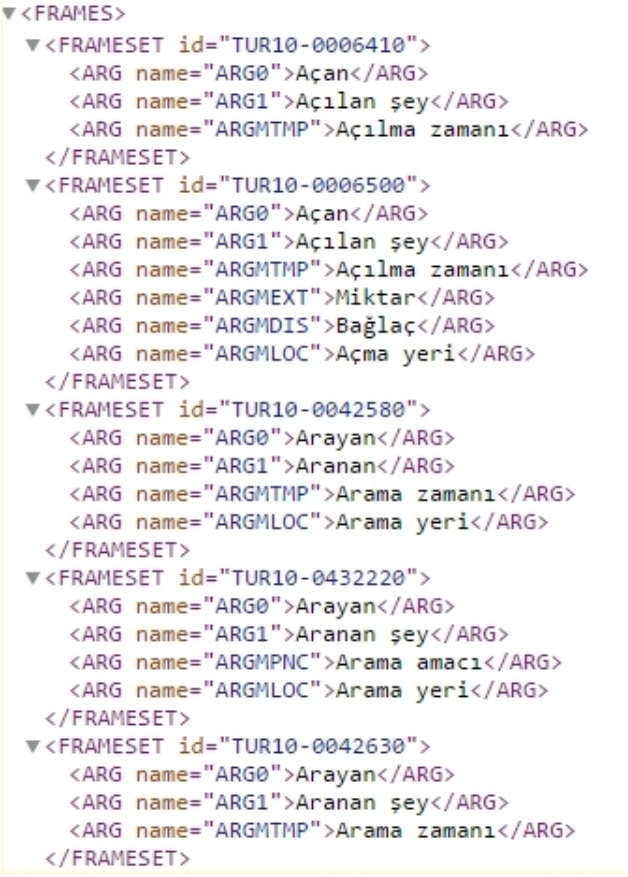


Figure 3.5: A part of the frame file : Different framesets with unique verb sense id are listed under FRAMES tag. In each frameset list of roles for that verb sense is stored.

Also, modifiers are extensively detected in the sentences with a total number of 1950 ArgMs are inserted into the framesets. The most frequent modifiers are ArgM-DIS, ArgM-LOC, ArgM-MNR and ArgM-TMP with more than 200 occurence.

### 3.2.2 Annotation Process

Once framesets are constructed, annotation of the translated corpora is straightforward. Annotation process is performed by the same team members. This time divided corpora is distributed in different order to avoid same person to complete both framing and annotation process. Thus, in that way any defect in the framing phase is revealed during the annotation phase. In such a case annotator request

Table 3.2: Argument statistics from the Frame file. Arg 0 and Arg1 are the most occured arguments as expected.

| Argument | # Occurrence |
|---|---|
| Arg0 | 1360 |
| Arg1 | 1289 |
| Arg2 | 142 |
| Arg3 | 10 |
| Arg4 | 4 |
| Arg5 | 1 |
| ArgM-CAU | 63 |
| ArgM-DIS | 336 |
| ArgM-DIR | 27 |
| ArgM-EXT | 154 |
| ArgM-LOC | 267 |
| ArgM-MNR | 319 |
| ArgM-ADV | 172 |
| ArgM-PNC | 127 |
| ArgM-TMP | 486 |

adding the necessary role to the roleset of the verb sense, hence annotation phase becomes a control for the framing phase.

Although PropBank annotation guidelines [37] provides a crucial basis for our guidelines, some of the arguments are omitted in our context. For some argument types such as ArgM-MOD, ArgM-NEG, mostly there is no equivalent word in Turkish translation instead this semantic information is concealed in the suffixes added to the verb. Figure 3.6 clarifies the situation, ArgM-MOD, ArgM-NEG arguments in English sentence are "will" and "not" words respectively. However, there are no equivalent words in the Turkish translation and the leaves of English words are absent. Instead, semantic information regarding these words are encapsulated in the suffixes of the predicate with "-me" and "-ecek" suffixes. Obviously, Turkish as an agglutinative language permits any given verb to use modal or negative suffixes. Also, English passive or causative word groups can be equivalent to a single Turkish verb which contains derivational affixes. These derived verbs are included as different verbs in our dictionary regardless of having the same semantic basis, so they are interpreted as different verbs in the current

architecture.

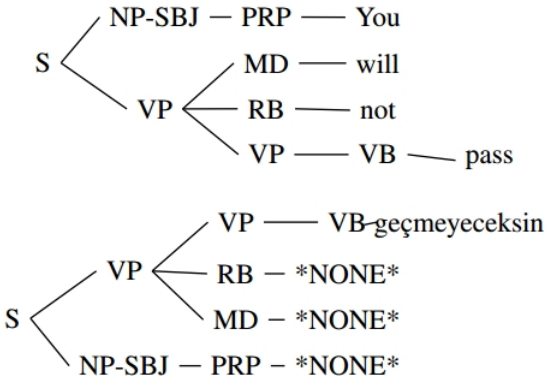


Figure 3.6: English parse tree and Translated Turkish counterpart: Modal and Negation words in the English sentence is represented as suffixes of the predicate in Turkish sentence.

### 3.2.3 Quality & Reliability of the Annotations

To ensure quality and reliability of the annotations an expert annotator outside the team annotated randomly selected 250 sentences. During the annotation, previously constructed framesets are given to the annotator. Also, she was allowed to use framing interface if an undefined roleset is required to annotate the sentences. Then annotations previously performed by the team are compared with the annotations of the expert. In the sample corpus, annotated sentences contain 2392 annotations. Out of these 2392 annotations, 1879 annotations matched with the expert annotator which corresponds to an 79% agreement.

### 3.2.4 Annotation Tool

As PropBank construction is spread out to many different languages, many different tools are proposed for frame composition and annotation. Available tools for PropBank annotation and frames file edition are Jubilee and Cornestone [41], [42], SALTO [58], TrEd [50] etc. Yet already, we are using an in-house NLP Toolkit

which supports all the operations mentioned in Turkish PropBank section above. Therefore we have integrated a PropBank editor to our toolkit to use the same infrastructure.

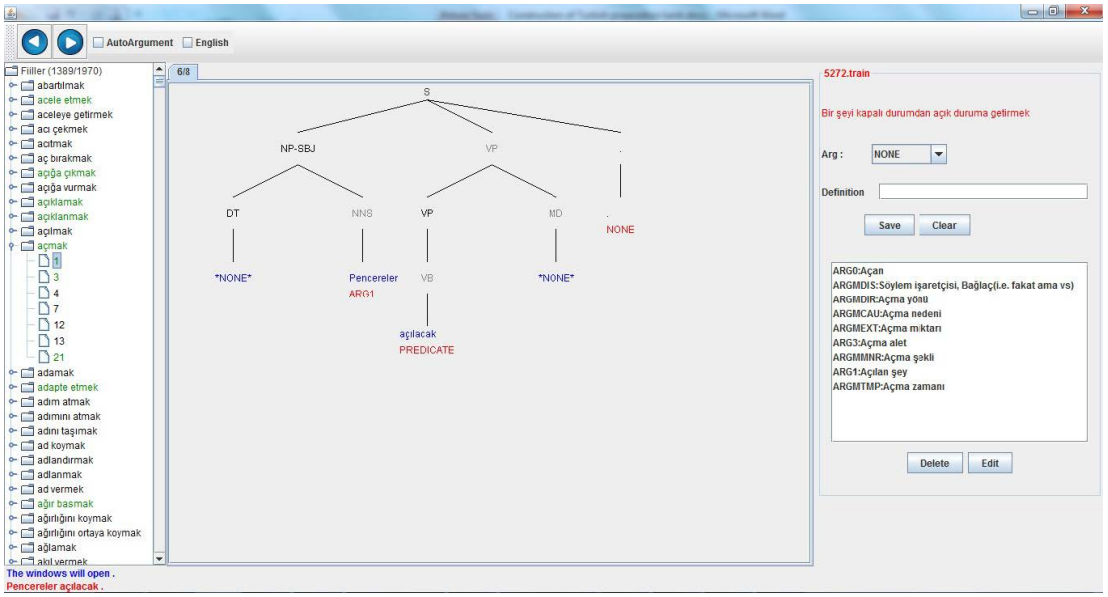


Figure 3.7: A screenshot of the PropBank semantic labeling tool : Verbs are coloured with respect to tagging status and listed in the left pane and sentences with the selected verb is positioned in the middle pane where annotator can tag each leaf in terms of PropBank roles. Role list for that verb is managed in the right pane.

A screenshot from the PropBank editor is presented in Figure 3.7. The list of verbs in the corpora is placed on the left side of the screen. Once a verb is clicked in the list, sense numbers are listed below. Sense selection triggers application to load first sentence tree to the middle and both English and Turkish sentence to the bar below. The data file name and sense explanation are also written above the sentence tree. In the framing phase, framers use this screen to examine all sentences of the verb sense by visiting all the trees via arrow buttons. Whenever a new role is found, it can be added to a frame file by filling the form in right panel. This panel supports framers to select the desired argument type in the dropdown menu above, semantic definition of the argument. After save button is clicked, argument record is added to the frameset of the verb sense. Framers can also delete or edit the argument record with the corresponding buttons. Figure 3.8 presents frame file insertion in detail.

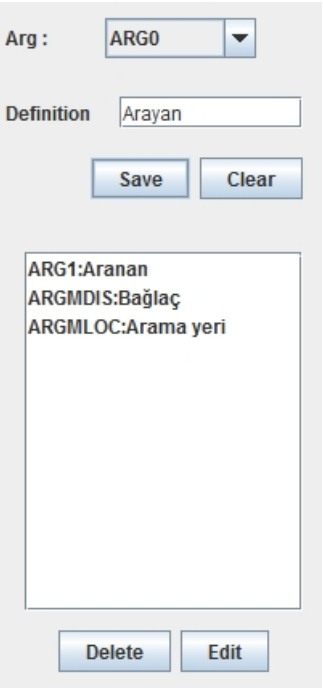


Figure 3.8: Argument insertion for the selected verb. Users first select the argument type then edit the definition field with the meaning of the argument for the selected verb

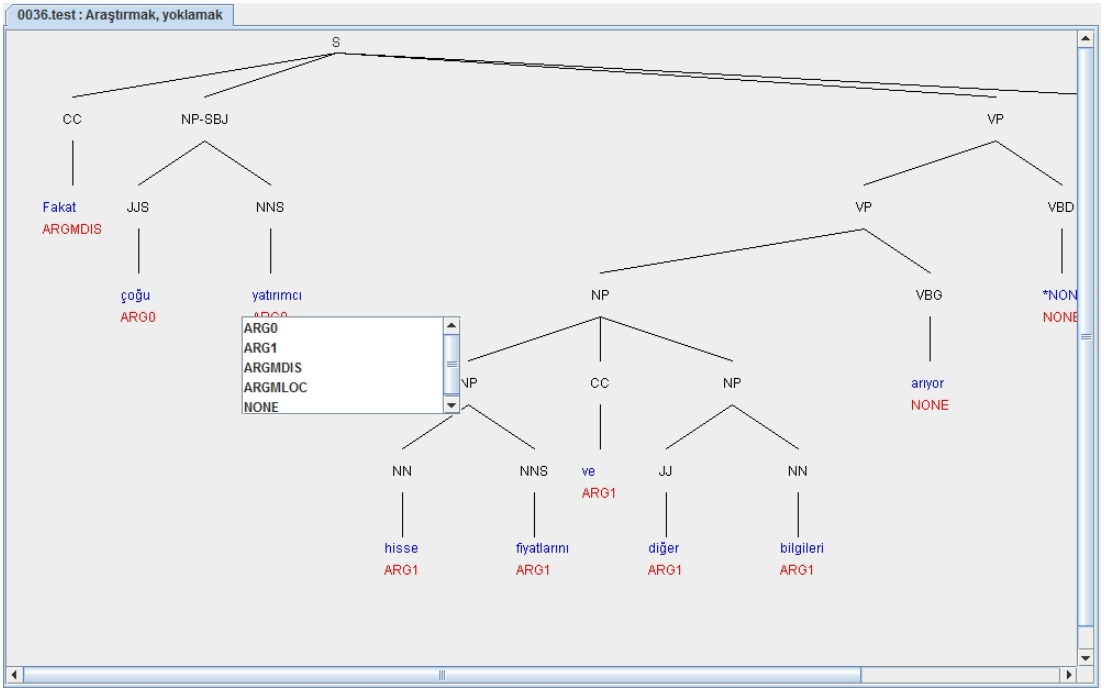


Figure 3.9: Annotation for the selected sentence is done by simple selecting the arguments of the predicate from the dropdown for each word.

When framer finishes all the sentences for a verb sense, rolesets are ready for the annotation. Annotation process again takes place in the same screen. Tree nodes in the middle area are clickable. Once the user clicks a node, role list that can be assigned to the node pops up. Figure 3.9 demonstrates annotation screen. After the selection made, selected role is printed below the node. In the same manner, annotator can visit all the sentences of the verb sense with the arrow buttons above.

Also, we have provide some functionalities to make annotation process even easier. Annotation of sentences is straight forward, but in some cases agent theme annotation is confusable. Sentences with unergative and unaccusative verbs require additional attention. As we previously stated for the passive sentences, subjects of unaccusative verbs should be tagged with Arg1. As an example, in the sentence "Gemi battı", "Gemi" is not Arg0. In addition, in some cases, the subject of the sentences becomes neither Arg0 nor Arg1 but an indirect object. To get rid of erroneous annotations, we also add warnings to the annotation interface for complex verb structures. Whenever an annotator attempts to annotate a complex verb instance, a warning sign with a description about the verb structure appears below the verb meaning, as in Figure 3.10. For the unergative and unaccusative distinction different diagnostics are explained in [73], we include these methods in Appendix A.

Another feature for simplifying annotation process; two check boxes are added to the top of the screen. "AutoArgument" check box, is used for predicting arguments in the sentence by evaluating the simple rules we have designed. This rules use a greedy approach for determining the argument tag. We simply detect nodes in the path from root to the leaf node where the word resides, and look to the syntactic tags for determining the word tag.

The second check box "English", shows the argument labels for the words in the original English sentence below the corresponding Turkish words. Since we translated the sentences from English PropBank, most of our sentences have

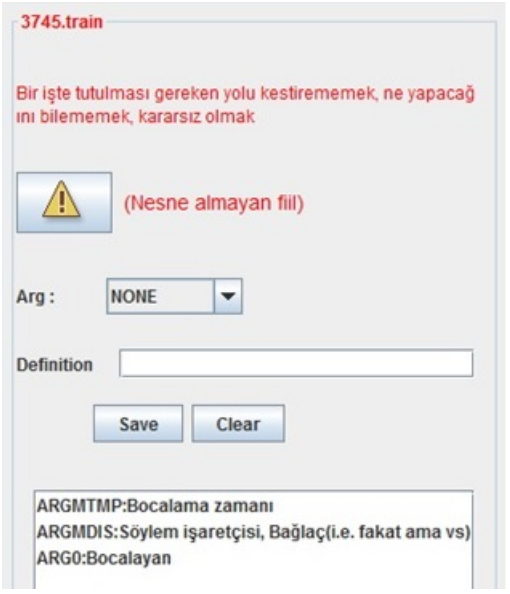


Figure 3.10: A warning about the verb structure is shown below the meaning.

equivalent annotated English sentence. These annotated sentences guide and help annotators to check their annotation. Figure 3.11, shows an example view where arguments of the original sentence are listed under Turkish translation.

Also, we have coloured the verb senses in the left pane. If there exists any verb sense that has no frame file, then it is coloured with red, so users first dive into that verb and add frame elements. When a verb sense has frame but not all instances

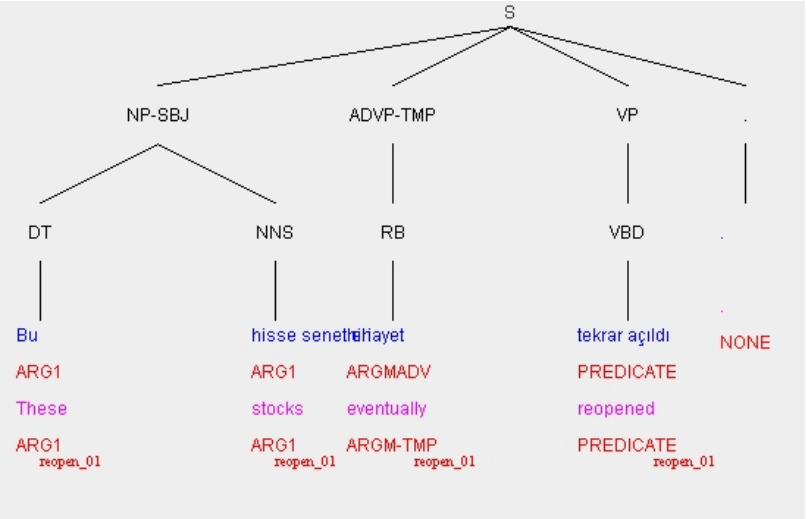


Figure 3.11: Tags for the English sentence are presented below the Turkish translation.

are tagged then it is coloured with green, and users understand annotation of the verb sense is in progress. Finally, if the verb sense is fully framed and annotated, it becomes black.

# Chapter 4

# Comparison of Turkish Proposition Banks

## 4.1 PropBank studies for Turkish language

Although PropBank studies are widespread among different languages, there exists only a couple of studies for Turkish language. Recently Şahin presented their study [28], [29] to create Turkish PropBank frames by incorporating Crowdsourcing techniques. They used Crowd intelligence to deal with verb sense annotation prior to the frame creation task. Then, Şahin and Adalı report the semantic role annotation [30] of arguments in the Turkish dependency treebank. They construct the proposition bank by using ITU-METU-Sabancı Treebank (IMST) [74] and later align it with IMST Universal Dependencies (UD). IMST is a syntactically-annotated corpus with sentences from Metu Turkish Corpus, which includes modern Turkish texts from 10 distinct genres. Şahin and Adalı frame 772 verbs and 1285 verb senses. They suggest using morpho-semantic features to calculate frame files for the verbs derived from verb roots by applying valency patterns. They use the Turkish root verbs provided by the Turkish Language Association (TDK) and the Turkish National Corpus (TNC).

In the frame creation process, they use Cornerstone [75], an open source Propbank frameset editor. They adjust Cornerstone for Turkish by adding two dropdown menus to supply case marking information and possible suffixes for Turkish verbs. Verb sense disambiguation of the frames and argument annotation are completed

by using crowd sourcing. 20060 semantic roles are annotated in 5635 sentences. Validity and quality control process of crowd sourcing techniques are discussed and results are evaluated in their study.

On the other hand, as presented in Section 3.2 we have constructed a Turkish Proposition Bank [31]. 9560 sentences containing a maximum of 15 tokens from the translated Penn Treebank II [71], [72] are used to generate the proposition bank thus the parallelism between our work and English PropBank gives us an opportunity to get enough corpus for manual annotation. Along with the annotated corpus, framesets are created for 1914 verb senses.

Another difference between our study and studies of Şahin and Adalı, we use an in-house-developed freely-available toolkit, NLP Toolkit, both for frame creation and annotation processes. This tool also includes other NLP methods, for preprocessing words in the dataset such as morphological and semantic analyses for verb sense selection. The tool is used by multiple annotators for frame creation and annotation of the words by hand. Frames are stored in a single file where each verb sense has a frameset. Framesets are distinguished by a unique synset id, e.g. "TUR10-0006410". Synset ids are taken from a Turkish WordNet based on the Contemporary Dictionary of Turkish provided by TDK. WordNet is a graph data structure where the nodes are word senses with their associated word forms and edges are semantic relations between the sense pairs.

## 4.2 Propbank Comparison

In order to compare framesets, a mapping between proposition banks is required. In the subsection 4.2.1, frameset mapping process is described, and then in subsection 4.2.2 comparison procedure is presented in detail.

### 4.2.1 Frame Matching

Turkish proposition banks discussed in Section 4.1 are built on different datasets, even frame file standards differ from each other. However, both of them use the verbs from the dictionary of Turkish Language Association. As we state in the previous section, root verbs of the first study are taken from TDK as in the second study where the synset ids are taken from WordNet based on the TDK dictionary. So we gathered frame files and WordNet to match verb senses. Şahin serves their frame files from the Turkish PropBank github website. We downloaded frame files and parsed each frame file to collect verb senses. In Figure 4.1, sample frame file for "bekle" is presented. This predicate has three senses stored in "roleset" tags. In each tag, the "name" attribute contains the meaning of the verb sense.

```
▼<frameset>
  ▼<predicate lemma="bekle">
    ▼<roleset id="bekle.01" name="Birini ya da birşeyi beklemek" vncls="">
      ▼<roles>
        ▼<role descr="bekleyen kişi" f="" n="0" suffix="NOM">
            <vnrole vncls="" vntheta="agent"/>
          </role>
        ▼<role descr="beklenilen kişi/şey" f="" n="1" suffix="ACC">
            <vnrole vncls="" vntheta="theme"/>
          </role>
          <note/>
        </roles>
      ▼<example name="tdk.01" src="" type="">
          <text>Ben de seni bekliyordum zaten.</text>
          <arg f="" n="0" suffix="">Ben de</arg>
          <arg f="" n="1" suffix="">seni</arg>
          <arg f="adv" n="m" suffix="">zaten</arg>
          <note/>
        </example>
        <note/>
      </roleset>
    ▶<roleset id="bekle.02" name="Ummak" vncls="">...</roleset>
    ▶<roleset id="bekle.03" name="Korumak">...</roleset>
      <note/>
    </predicate>
    <note/>
</frameset>
```

Figure 4.1: Frame file for "bekle" taken from Turkish PropBank github web page.

Likewise, we downloaded the WordNet file from the website [76] and the frame files from the website [31] for our proposition bank. Since our rolesets use synset ids from WordNet, it is sufficient to use WordNet to match verb senses. In Figure

4.2, a part of a frame file is presented for “bekle” where WordNet ids are stored in the “id” attribute.

```
<FRAMESET id="TUR10-0089650">
    <ARG name="ARG1">Beklenen</ARG>
    <ARG name="ARG0">Bekleyen</ARG>
    <ARG name="ARGMTMP">Bekleme zamanı</ARG>
</FRAMESET>
<FRAMESET id="TUR10-0089780">
    <ARG name="ARG1">Beklenen</ARG>
    <ARG name="ARGMDIS">Söylem işaretçisi</ARG>
    <ARG name="ARG2">Beklenilen şey/kişi</ARG>
    <ARG name="ARGMTMP">Beklenme zamanı</ARG>
</FRAMESET>
```

Figure 4.2: Frame file of our proposition bank [31].

WordNet entry for “TUR10-0089560” is shown in Figure 4.3. The fourth sense of “Beklemek” in the WordNet has the same meaning as bekle.02 in the Turkish PropBank of Şahin.

```
<SYNSET>
    <ID>TUR10-0089650</ID>
    <SYNONYM>
        <LITERAL>beklemek<SENSE>4</SENSE></LITERAL>
        <LITERAL>demek<SENSE>8</SENSE></LITERAL>
        <LITERAL>umut etmek<SENSE>1</SENSE></LITERAL>
    </SYNONYM>
    <POS>v</POS>
    <ILR>ENG31-00721987-v<TYPE>SYNONYM</TYPE></ILR>
    <DEF>Ummak</DEF>
    <EXAMPLE>Nikâhtan bu kadar keramet bekleme!</EXAMPLE>
</SYNSET>
```

Figure 4.3: WordNet synset for Frameset id “TUR10-0089560”.

We automated this process to match verb senses across the proposition banks. The frame matching method simply creates a list of verb sense ids along with the corresponding meanings for both proposition banks. If there is any verb sense with the exactly same meaning, our method pairs the ids at the top for this verb. For unmatched verb senses, meanings are listed for both sides. After we generated the verb sense lists, linguists in the team matched the verb senses manually. For instance, for the verb bağışlamak, the linguists were provided with a list as in

Table 4.1: Before the matching

| Verb | Example sentence | PropBank Senses | WordNet Senses |
|---|---|---|---|
| bağışlamak | çocuk elindeki çiçek demetini kumandanın ayağı altına atarak, babamı bağışlayınız, diyordu. | Hoş görmek, affetmek | Herhangi bir kötü davranış için ceza vermekten vazgeçmek; affetmek |
| | ödünç aldığı parayı bile kendinden daha ihtiyaçlısına bağışlayan ancak bir masal adamıdır. | Bir mal veya hakkı karşılık beklemeden birine vermek | Görevden, işten çekmek |
| | | | Deyimlerde "Tanrı esirgesin, ayırmasın" anlamlarında kullanılan bir söz |
| | | | Bir mal veya hakkı karşılık beklemeden birine vermek, teberru etmek |

Table 4.2: After the matching

| Verb | Example sentence | PropBank Senses | WordNet Senses |
|---|---|---|---|
| bağışlamak | çocuk elindeki çiçek demetini kumandanın ayağı altına atarak, babamı bağışlayınız, diyordu. | Hoş görmek, affetmek | Herhangi bir kötü davranış için ceza vermekten vazgeçmek; affetmek |
| | ödünç aldığı parayı bile kendinden daha ihtiyaçlısına bağışlayan ancak bir masal adamıdır. | Bir mal veya hakkı karşılık beklemeden birine vermek | Bir mal veya hakkı karşılık beklemeden birine vermek, teberru etmek |

Table 4.1 including two senses from Şahin's PropBank and four from WordNet. Additionally, two exemplary sentences were given to make the distinction among the senses clearer. After analyzing the senses in PropBanks, the corresponding senses were matched as it can be seen in Table 4.2.

### 4.2.2 Frame Comparison

After matching proposition bank of Şahin and WordNet, we composed a mapping file which consists of a WordNet id and an equivalent roleset id as in Table 4.3. As we previously stated in Subsection 4.2.1 and Figure 4.1, the "id" attribute of "roleset" tag contains the necessary information to find the corresponding WordNet Id to match the rolesets of the two Turkish proposition banks. So, for each frame file of Şahin, we listed rolesets and found the corresponding WordNet ids from the mapping. Then, we searched for the frameset of the WordNet id in our frame file. Once we found a frameset, we compared the arguments one by one.

Table 4.3: Mapping between WordNet and PropBank (Şahin)

| RolesetId | WordNetId |
|---|---|
| aban.01 | TUR10-0000360 |
| abartıl.01 | TUR10-0000500 |
| abart.01 | TUR10-0129480 |
| acık.01 | TUR10-0002820 |
| acı.01 | TUR10-0002890 |

## 4.3 Results

After examining 1285 senses of 772 verbs in total, it has been observed that the majority of the senses (1111 senses of 711 verbs) in the PropBank(Şahin) and WordNet do match. Although both datasets were created by using the items in TDK (the Turkish Language Association), there are differences between them. The reason behind those differences can be that they were created by using different versions of TDK.

The first difference is that 170 senses in Şahin's PropBank dataset do not have matching senses in WordNet. Apart from these senses, there were also 4 frame files created for the suffixes "-da", "-la", "-lan", and "-laş", which add new roles

to the verbs that they produce. We also omitted these frames since they have multiple correspondents in WordNet.

The second difference is that 4 verb senses in WordNet were merged in Şahin's frame files and thus, one-to-one match between those senses was not possible. For example, for the verb bulmak (to find), one of the senses provided in Şahin's frame files was keşfetmek, icat etmek (to discover, to invent). However, this sense was split into two in WordNet: Varlığı bilinmeyen bir şeyi ortaya çıkarmak—Var olduğu bilinmeyen bir şeyi bulmak (to discover) and İlk kez yeni bir şey yaratmak (to invent). The other three verbs whose senses were merged are bulunmak, hırpalamak, and utanmak. For these verb senses, one of the matching WordNet sense is selected.

In frame and roleset comparison, we have tried to compare the roles of 1111 verb senses that are mapped to WordNet. However, the rolesets for 519 verb senses do not exist in our frame file, which is equal to half of the verb senses from the intersection of WordNet and frames of Şahin. So we have only compared 592 common verb senses in this step. On the other hand, there are 1322 verb senses in our frame files which do not exist in Şahin's PropBank and therefore, are not included in the comparison. The relation between two proposition bank is presented in Figure 4.4.


Figure 4.4: Intersection of verb senses between Şahin's and our frame files. 519 verb sense occur only in Şahin's frame files, 1322 more verb senses are listed in our frame file and 592 verb senses match in both resource.

When we process the rolesets, we have observed that the rolesets of Şahin's frames have 2713 roles for 1285 verb senses where 372 roles are added for the 170 verb senses which do not exist in WordNet. Also, 1050 roles are added for 519 verb senses not exist in our frames. In the matching frame files, there are 430 roles that exist in Şahin but not exist in our frames. On the other hand, we offer 1568 roles for 592 verb senses in common. Within this common set, 861 roles are the same with the roles of Şahin. Remaining 707 roles do not exist in Şahin's frames. When we investigate further, we have found out that access modifiers such as temporal, locative, purpose, and cause are added along with the ARG0-ARG5 arguments to the rolesets in our study. In Şahin's frames, these modifier roles are not included, they only add the arguments from ARG0 to ARG5. Table 4.4 lists number of roles that are same in both proposition bank. In Table 4.5 number of roles that exist in Şahin but not exist in our frames displayed. And in Table 4.6 number of roles that exist in our frames but not exist in Şahin presented. As you can see, ARG0 and ARG1 arguments are the most common roles that are same in both propbanks as expected. The main difference for propbanks is access modifiers treated differently in both propbank.

Table 4.4: Number of same roles in the roleset

| Argument | Count |
|---|---|
| ARG0 | 404 |
| ARG1 | 441 |
| ARG2 | 16 |

Table 4.5: Number of roles in Şahin differs from our study.

| Argument | Count |
|---|---|
| ARG0 | 86 |
| ARG1 | 90 |
| ARG2 | 158 |
| ARG3 | 48 |
| ARG4 | 44 |
| ARG5 | 1 |
| ARGm | 3 |

Table 4.6: Number of roles in our frames differs from Şahin.

| Argument | Count |
|---|---|
| ARG0 | 82 |
| ARG1 | 39 |
| ARG2 | 24 |
| ARG3 | 8 |
| ARGMADV | 36 |
| ARGMCAU | 14 |
| ARGMDIR | 6 |
| ARGMDIS | 90 |
| ARGMEXT | 54 |
| ARGMLOC | 80 |
| ARGMMNR | 88 |
| ARGMPNC | 41 |
| ARGMTMP | 145 |

We present the comparison and the matching process of the newly-created Turkish Proposition Banks. As Turkish is not very rich in terms of linguistic resources, relating these two propbank studies gives the opportunity to improve both datasets. Frame files that do not exist in both sides can be imported or role differences for matched verb senses can be reevaluated.

There are 1914 rolesets and 4757 roles for these rolesets in the our study and Şahin's frame files consist of 1285 rolesets and 2713 roles defined in frame files. Also, mapping these proposition banks with WordNet is important. Turkish WordNet used for the mapping is aligned with the English counterpart, which may enable the transfer of linguistic information from other proposition banks and give possibility to extend datasets.

# Chapter 5

# Automatic Turkish PropBank Generation

## 5.1 Automatic Turkish PropBank using Parallel Sentence Trees

As we previously stated in Section 3.1, Penn Treebank structure offers advantages for building fully tagged data set in accordance with syntactic labels, morphological labels and parallel sentences. We used this structure to add English PropBank labels for each word in the corpus. In this manner, we exploited our parallel dataset to transfer English PropBank annotations to an automatic Turkish PropBank.

### 5.1.1 English PropBank Labels

Original English PropBank corpus [77] is accessible through Linguistic Data Consortium (LDC). This resource is the initial version of the English PropBank and it only includes the relations with verbal predicates. In the newer versions adjective and noun relations are also annotated. Since we compare projection results with our manually annotated corpus [31] which only contains verbal relations, we use the initial version of the English PropBank. We downloaded this dataset and imported annotations for the selected sentences. After this step 6060 sentences among 9558 were enhanced with the English annotations. Below in Figure 5.1, a sample sentence is presented. English annotations are inserted inside "english-Propbank" tags right after Turkish annotations which reside in "propbank" tags.

Some of the words have only english annotation, because there is no word translated in the Turkish sentence for this node. As an example, "their" in Figure 5.1 has annotations in the englishPropbank tag but there is no equivalent translation in Turkish, presented as "*NONE*", so propbank tag does not exist. English tags have predicate information that annotation belongs to. "Müşterilerinin" in the same example has "ARG0$like_01#ARG1$think_01" in the englishPropbank tag which means there exists at least two words whose root is in verb form. Here the word is annotated with respect to "like" and "think" separately. "englishPropbank" tags separate multiple annotations with "#" sign and in each annotation predicate label and role is distinguished by "$" sign where in the propBank tag we used WordNet id of the predicate for the annotation. Argument role is separated by the "$" sign. As for Turkish PropBank, we only cover the predicate for annotation at the moment.

```
(S
(NP-SBJ
(NNS{morphologicalAnalysis=müşteri+NOUN+A3PL+P3PL+GEN}
{metaMorphemes=müşteri+lArH+nHn}
{turkish=Müşterilerinin}
{english=customers}
{semantics=TUR10-0565710}
{namedEntity=NONE}
{propBank=ARG1$TUR10-0231190}
{englishPropbank=ARG0$like_01#ARG1$think_01}
{englishSemantics=ENG31-10004189-n})
(PRP$ {morphologicalAnalysis=*NONE*}
{metaMorphemes=*NONE*}
{metaMorphemesMoved=nHn}
{turkish=*NONE*}
{english=their}
{englishPropbank=ARG0$like_01#ARG1$think_01}))
```

Figure 5.1: Part of a sentence tree : English PropBank annotations reside in "englishPropBank" tags.

### 5.1.2 Transfering Annotations to Automatic Turkish PropBank using Parallel Sentences

After importing English annotations, it is necessary to determine predicate(s) of the Turkish sentences. In most cases, Turkish predicates appear at the end

of the sentence. So, we reversed the node list. Morphological structures of the words in the list are examined to detect predicate candidates. All the words morphologically and semantically analyzed in translated Penn TreeBank. We have used "morphologicalAnalysis" tag to check the morphological structure of the words. In Figure 5.1, sample morphological structure is displayed.

The word which have a verb root and verb according to last inflectional group is treated as the predicate of the sentence. Once we found a word suitable for these conditions, we gathered English PropBank annotation. If it is also labeled as predicate in English proposition bank, we got the predicate label, e.g. like_01, to find annotations with respect to this predicate. As previously stated in Section 5.1.1, English labels have multiple annotations for each verb label in the sentence. We searched for the found predicate label in the annotations and transfered annotations matching with the predicate label. If we could not find a predicate in Turkish sentence or the corresponding English label did not contain Predicate role annotation, we skipped to the next predicate candidate.

During the transfer, a mapping was needed due to the difference between English and Turkish [31] argument labeling. English PropBank corpus has "-" sign in ArgM's like ARGM-TMP and also some of the arguments from Arg1 to Arg5 are labeled with the prepositions such as ARG1-AT, ARG2-BY etc. We processed these differences and then transferred labels into the "propbank" tags. After analyzing Turkish sentences we found out some sentences have more than one predicate, so we continued to search for another predicates in the sentence and ran the same procedure for each predicate candidate.

### 5.1.3 Results & Evaluation

Annotations gathered from the Section 5.1.2 were compared with the Turkish hand-annotated proposition bank [31]. Comparisons were done at the word level by checking the annotations for each corpus. We only considered sentences whose

annotations were transferred from English PropBank. So, 6060 parallel hand-annotated sentences were taken into account. Among these 6060 sentences, 11 sentence do not have predicate annotation. Also, when we look at Turkish counterparts of the remaining 6049 sentence, 837 sentences did not have a verbal predicate in hand-annotated proposition bank. Therefore, in 5212 sentences, 44779 word annotations were compared. 31813 annotations were transferred from English to Turkish. Results of the comparison are presented in Table 5.1.

Table 5.1: Results of the comparison between automatic proposition bank and hand annotated (HA) proposition bank.

| **Transferred** | | **Untransferred** | |
|---|---|---|---|
| Correct | 19373 | Invalid HA | 8837 |
| Incorrect | 6441 | Valid HA | 4129 |
| Undetermined | 5999 | | |
| **Total** | 31813 | **Total** | 12966 |

- 19373 words annotated with PropBank roles correctly .
- 6441 annotations are incorrect, PropBank tags are different in both corpus.
- 5999 annotations are undetermined: 5396 word not annotated, 603 NONE tags exists in hand annotated corpus while valid PropBank labels transferred from English annotations. Annotations to be compared is not valid so we did not include this set in the evaluation.

When we remove undetermined 5999 words in the comparison; 19373 annotations from 25814 annotations are correct, which gives us ~75% accuracy for transferred and comparable set. These 5999 annotations may be hand-annotated and recompared for validity of the transferred annotations.

For the erroneous transfers, rules may be integrated during the transfer of English annotations. Especially, annotations for passive, unergative, and unaccusative verbs should be processed since subjects of the sentences are not Arg0. In the projection phase, the preposition information on English PropBank labels can

be used to determine transferred role. We can also control English and Turkish framesets and compose mappings between roles for each predicate. Moreover, there exists differences on ArgM usage. Hand-annotated corpus is mainly tagged with ArgM's instead of having numbered arguments. Again frame files may give clues for the roles Arg2 to Arg5 for mapping to which adjunct role.

In Table 5.2, we present occurrences of erroneous annotation transfers. Only top ten occurrences are presented. Arg0-Arg1 transfers are the most occurred incorrect transfers 1843 among 6441 incorrect annotations. Second most occurred error is in Arg1-Arg2 labels. Errors in Arg0-Arg1 and Arg1-Arg2 labels forms ~44% of the transfer errors.

Table 5.2: Counts of different argument annotations between transferred annotations and hand annotations.

| **Different Arguments** | **# of Occurrence** |
|---|---|
| ARG0-ARG1 | 1843 |
| ARG1-ARG2 | 961 |
| ARG2-ARGMEXT | 462 |
| ARG1-PREDICATE | 255 |
| ARG0-ARG2 | 229 |
| ARG4-ARGMEXT | 226 |
| ARG1-ARGMPNC | 220 |
| ARG1-ARGMMNR | 186 |
| ARG1-ARGMTMP | 160 |
| ARG1-ARGMLOC | 148 |

On the other hand, when we look at the all word results, 12966 roles were not transferred. If we take these untransferred instances as incorrect; 19373 annotations are correct, 6441 incorrect, 12966 not transferred; which means that 19373 annotations out of 38780 annotation are true and the accuracy drops to ~50%. However, 8837 of untransferred annotation are not valid in the hand-annotated corpus. Only 4129 are valid PropBank arguments in the hand-annotated dataset. In this respect, if we count only valid arguments for untransferred annotations, accuracy is ~65%.

When we examined untransferred annotations, we found out our predicate prediction method failed to identify words labeled as predicate in the hand annotated corpus. We identified 5061 predicates during the annotation transfer whereas there exist 5491 predicate annotation in the hand annotated corpus. Among these 5061 predicates 4927 words annotated as predicate in the hand annotated corpus too. 134 words are predicted as predicate but are not labeled as predicate in the hand annotated corpus.

Implemented method generates automatic Turkish proposition bank by transferring cross-language semantic information. Using the parallelism with English proposition bank gives us an opportunity to create a proposition bank in a short time with less effort . Preliminary results of the automatic proposition bank are supportive. We currently have 65% accuracy with the hand-annotated proposition bank [31]. When we consider only transferred annotations, accuracy is rising to ~75%. In this method, annotations transferred using parallel tree structures of the corpus. However, having such a parallel resource in tree format is difficult. Annotations of the semantic roles can be implemented using shallow parse similarity. Also, we did not transfer parallel frames from English proposition bank. While transferring annotations, we can map predicates and generate frames automatically. The current set up can be also used for different languages if parallel sentence trees provided to test validity of the approach.

## 5.2 Automatic Turkish PropBank Using Parallel Sentence Phrases

In the previous section, annotation projection using parallel sentence trees is discussed. However, finding such a resource in a special format is difficult especially if you are working with a resource-scarce language. Most of the time creating a formatted parallel resource like tree structured sentences complicates translation procedure. In this section, automatic generation with only translated sentences will be examined.

### 5.2.1 Phrase Sentence Structure

In Section 3.1, we used the translated Penn Treebank II [71], [72] sentences with tree structure. For the phrase sentences, English sentences re-translated without tree structure. Prior the annotation projection, linguists in the team annotated phrase sentences and populated "propbank" and "shallowParse" tags so that we check the correctness of the annotation transfer. 6511 sentences among 9557 phrase sentences have predicate according to hand annotations. However, only 5259 sentences have English PropBank annotation, so we take this set to transfer annotations. As you remember, the same number in the previous section was 5212. Here translation and annotation differences change the processed sentence count.

Tag structure of Penn Treebank is preserved to simplify morphologic and semantic analysis requirements during the annotation transfer. In Figure 5.2, sample phrase sentence can be seen. Unlikely Figure 5.1, syntactic tags which indicate tree structure, explained in Section 2.1.3, are not included. We used original tree formatted English sentence to extract English propbank annotations. However, since the target sentence do not have tree structure definition we used other word alignment methods to determine annotation projection.

### 5.2.2 Semantic Alignment Using WordNet

In order to transfer annotations, first we tried to match predicates of English sentence and Turkish translation. Again we utilize "morphologicalAnalysis" tags to determine predicate candidates in the phrase sentence. Words which have a verb root and verb according to last inflectional group is treated as the predicate candidates of the sentence. Once we found all the words ensures these conditions, we gathered all English PropBank annotation labels which are tagged as "Predicate" in 'englishPropbank" tag. To align predicates in different languages, we tried to


Figure 5.2: Part of a phrase sentence : Translated words in turkish tags. Helper tags gives additional information for each word.

exploit WordNet's interlingual mapping capabilities. For each predicate in English sentence we find Turkish translation by searching English synset id in the WordNet. English synset id is located in englishSemantics tags as in the sample in Figure 5.1. If there exists any translation in the WordNet, we take Turkish synset id and search it in the predicate candidates found for phrase sentence. Whenever translation found, we align predicates and try to transfer annotation with respect to aligned English label. For annotation transfer of other arguments we again align words using WordNet's interlingual mapping. An example WordNet record is presented in Figure 5.3.

First results gathered with only WordNet mapping were very low. True annotation count is 2195 among 29168 annotations tagged manually which yields 7.53%. The rest of the annotations are not transferred and count as false. However, transferred false annotation count is only 342. When we examined further, we saw system heavily relies on semantic annotations for both English and Turkish words where some of the words failed to have semantic annotation. We look deeper into English words, 11006 words having valid arguments but do not have

```
<SYNSET>
    <ID>TUR10-0682580</ID>
    <SYNONYM>
        <LITERAL>sevmek<SENSE>5</SENSE></LITERAL>
    </SYNONYM>
    <POS>v</POS>
    <ILR>ENG31-01779085-v<TYPE>SYNONYM</TYPE></ILR>
    <ILR>ENG31-01779456-v<TYPE>SYNONYM</TYPE></ILR>
    <ILR>ENG31-01780873-v<TYPE>SYNONYM</TYPE></ILR>
    <ILR>ENG31-01781131-v<TYPE>SYNONYM</TYPE></ILR>
    <DEF>Yerini, şartlarını uygun bulmak</DEF>
    <EXAMPLE>Bu ağaç nemli ortamı sever.</EXAMPLE>
</SYNSET>
```

Figure 5.3: Sample WordNet record found by searching "ENG31-01781131-v", English synset id, from the sentence in Figure 5.1.

semantic annotation so we failed to match these words with Turkish counterparts. Addition to these words, 336 words in English tagged as Predicate, do not have semantic annotation which prevent us transferring annotations related with these predicates. We also observed 570 predicate candidates do not have mapping in WordNet. 4450 predicate candidate has mapping in WordNet but not matched with the words in Turkish sentence. Our method has predicted only 1342 true and 45 false predicate among 6716 predicates in 5259 files. For these 1387 predicates, there exist 9604 English words to be transferred. 3424 of the words do not have semantic annotation, and 729 words do not have mapping in WordNet. 2344 words have mapping but not matched with the words in Turkish sentence.

Some words are not annotated semantically such as, proper nouns, time, date, numbers, ordinal numbers, percentiles, fractional numbers, number intervals, and reel numbers. Most of these words are same in Turkish translation so we decide to match English and Turkish words by string match. For example if a sentence contains proper noun "Dow Jones", the same string also exists in the Turkish translation too. However, it may take additional suffixes so we tried to match words by looking root for of the Turkish word in the English word if English word does not have semantic annotation. Since most of the time, proper nouns, numbers, date - time words take suffixes in Turkish, we only check whether English

words starts with Turkish root word. Also, translational differences are encountered like decimal separator in English is "." where some Turkish translations "," is used. We replace this differences by looking whether the first morphological tag is "NUM". After these tunings, we rerun the procedure and get 2680 true and 531 false annotations which increases true annotations to 9.19%. Another problem is erroneous semantic annotations. If English and Turkish semantic annotation is not right, alignment is not possible. Even in the best scenario where both word is annotated, if WordNet mapping is incomplete, an alignment can not be established.

As an alternative we decided to reinforce annotation transfer by using constituent boundaries identified with shallowParse tags. Example of shallowParse tags can be seen in Figure 5.2. Prior to the annotation transfer, phrase sentences are annotated for constituent boundaries which can be used to group argument roles in the sentence. After transferring annotations with respect to semantic annotations, we run another method over phrase sentences which calculates maximum argument types for each constituent and tags any untagged word with the calculated max argument role within the constituent boundary. This procedure further enhance true annotations to 4255 but also increase false annotations to 1202. After constituent boundary calculation, correct annotation transfer percent is increased to ~14.59%. In Figure 5.4 annotation of the sentence 7076.train is presented. Untagged words in "Özne" and "Zarf Tümleci" constituent boundaries are tagged with the found argument role within the boundary.

### 5.2.3 Word Alignment Using IBM Alignment Models

Word alignment through semantic relation requires fair semantic annotation for both languages and also sufficient semantic mapping between languages. We search different word alignment methods between English and Turkish sentences. IBM alignment models offer solution to our word alignment problem. IBM Models are mainly used for statistical machine translation to train a translation model

(8) [Daha az sıkı türden bir Senato versiyonu]$_{\text{Özne}}$ [aşağı yukarı beş yıl için]$_{\text{Zarf Tümleci}}$ [düşülebilirliği]$_{\text{Nesne}}$ [ertelerdi.]$_{\text{Yüklem}}$

(9) [Daha az sıkı türden bir]$_{\text{NONE}}$ [Senato]$_{\text{ARG0}}$ [versiyonu]$_{\text{NONE}}$ [aşağı yukarı]$_{\text{ARG2}}$ [beş]$_{\text{NONE}}$ [yıl]$_{\text{ARG2}}$ [için düşülebilirliği]$_{\text{NONE}}$ [ertelerdi.]$_{\text{PREDICATE}}$

(10) [Daha az sıkı türden bir Senato versiyonu]$_{\text{ARG0}}$ [aşağı yukarı beş yıl için]$_{\text{ARG2}}$ [düşülebilirliği]$_{\text{NONE}}$ [ertelerdi.]$_{\text{PREDICATE}}$

Figure 5.4: Annotation reinforced with respect to constituent boundaries: (8) constituent boundaries identified with shallowParse tags for sentence in 7076.train, (9) Argument roles for the same sentence after annotation transfer, (10) Argument roles for the same sentence after reinforce method run.

and an alignment model. IBM Model 1 [78] is the primary word alignment model offered by IBM. It is widely used for solving word alignments while working with parallel corpora. It is a generative probabilistic model that calculates probabilities for each word alignment from source sentence to target sentence. It takes a corpus of paired sentences from two languages as training data. These paired sentences are possible translation of the sentences from source language to target. With this training corpus, parameters of the model estimated using EM (expectation maximization). IBM Model 1 is limited for complex situation like many-to-one alignment, distortion (contiguous source words aligned to contiguous target words), and fertility (source words may generate a specific number of target words or dropped at all). IBM Model 2 has an additional model for alignment and introduce alignment distortion parameters. We decided to use IBM model 1 & 2 to establish word alignments instead of WordNet's interligual mapping. We input sentence pairs and gather alignment probabilities for each English word to Turkish equivalent. 244024 word pairs are taken as output where for each English word, 10 most probable Turkish words are listed. We insert these pairs into a database table to further investigate the word alignments. Alignment probabilities for word "Reserve" is presented in Table 5.3 and 5.4 for IBM Model 1 and 2 respectively.

After gathering alignment data, we transfer annotations to phrase sentences from

Table 5.3: Word alignment probabilities for English word "Reserve" calculated by IBM Model 1.

| English Word | Turkish Word | Probability |
|---|---|---|
| Reserve | Reserve | 0.722709170447948 |
| Reserve | Rezerv | 0.153284144983192 |
| Reserve | mevduat | 0.030562931419084 |
| Reserve | Bankası'nın | 0.027316643368162 |
| Reserve | kuruluşlarındaki | 0.013753319138588 |
| Reserve | komisyonları | 0.013753319138489 |
| Reserve | Bankasının | 0.006112586283128 |
| Reserve | kuruluşlarında | 0.004584439712863 |
| Reserve | komisyon | 0.004584439712863 |
| Reserve | Federe | 0.004584439712863 |

Table 5.4: Word alignment probabilities for English word "Reserve" calculated by IBM Model 2.

| English Word | Turkish Word | Probability |
|---|---|---|
| Reserve | Reserve | 0.677007554711816 |
| Reserve | Rezerv | 0.143607663120688 |
| Reserve | Federe | 0.061546141337438 |
| Reserve | Bankası | 0.052659720643273 |
| Reserve | tasarruf | 0.030721823477854 |
| Reserve | kuruluşlarına | 0.021173938324081 |
| Reserve | üzerindeki | 0.011118563367315 |
| Reserve | bu | 0.002120046139771 |
| Reserve | kurumlarına | 0.000044524905565 |
| Reserve | Merkez | 0.000000015437643 |

English PropBank labels in the tree structured sentences. For each phrase sentence first we reset all arguments and read corresponding English sentence. All words tagged with "PREDICATE" tag in English are stored into a map which includes predicate label from the "englishPropbank" tag *e.g.* "like_01" and english word from the "english" tag *e.g.* "like". Then we search alignments for each found English predicate. Here we observed that aligned Turkish words may not occur in the phrase sentence as they found in the alignment table. Words may include additional suffixes, so we use Finite State Machine(FSM) morphological analyzer available in our NLP Toolkit to extract roots of the aligned Turkish words. Since we have several possible morphological parse for each aligned word,

we created an array for possible roots. In parallel, we found predicate candidates from the phrase sentence as we stated in the previous methods. Then we tried to match aligned words and possible roots with the found predicate candidates. If there exists a predicate candidate that matches with the aligned word or one of its roots in the array, we tagged the candidate as "PREDICATE" and update map as predicate label and synset id of Turkish predicate.

After finishing predicate discovery, we tried to transfer annotations for found predicates. To do that we look for the annotations with respect to the predicate labels in the map. For each record in map we took the predicate label and corresponding Turkish synset id. When we found an annotation with this predicate label, first we extract the argument and try to find aligned word for the processed English word. In this phase we only transfer valid arguments except "PREDICATE". Predicates are only processed in the predicate prediction phase and if we encounter with a predicate afterwards, we pass and do not transfer. For the alignment again we find the most probable word from the table and use FSM morphological analyzer to extract possible roots. Then for each word we search Turkish sentence to match words with aligned word or possible roots extracted. If matched Turkish words do not have argument annotation, we transfer argument with the synset id found in the map record.

As we discuss in the previous annotation transfer procedure 5.2.2, some of the English words such as proper nouns, time, date, numbers, ordinal numbers, percentiles, fractional numbers, number intervals, and reel numbers stay same or take additional suffixes in Turkish translation. So we include the same method used for matching these words. In a case words are not aligned with the information from alignment table, and a valid annotation present in English word, we search exact string match or any word starts with the root of English word in the Turkish sentence.

In order to increase correct annotation we check the steps and try to fix any deficiencies. Annotation transfer is adversely affected by two inadequate process;

predicate prediction and erroneous word alignment. So we first inspect any drawback in predicate prediction. We list the words annotated with "PREDICATE" tag in hand annotated corpus and extract words could not be found by our prediction method. Two cases are liable for incorrect predicate detection. We present these cases in Figure 5.5. First one is the wrong morphological annotations for predicates. In example 11 available morphological analyses for "ödeyecekler" is shown. Most of the missing predicates have future tense suffix "-AcAk" (FUT) or ability suffix "-bil" (ABLE) where morphological analysis should be selected as verb for the last inflection. However, morphological analysis of some words with these suffixes is selected as noun or adjective. Since our method looks for words which are verbs in last inflection, these predicates can not be found. Also, there exist multiple word predicates where morphological analysis for individual words is not suitable for our verb constraint. In example 12 we show morphological analysis of "mutabakata vardı". Although "vardı" has eligible analysis for predicate prediction, "mutabakata" is not selected as predicate since it is noun. These words are tagged with "PREDICATE" tag in hand annotated corpus so number found predicates is descreasing.

(11) Morphological analyses available for ödeyecekler:
öde+VERB+POS+FUT+A3PL (Correct analysis for predicate)
öde+VERB+POSˆDB+NOUN+FUTPART+A3PL+PNON+NOM
(Incorrect analysis for predicate)

(12) Morphological structure for "mutabakata vardı":
mutabakat+NOUN+A3SG+PNON+DAT var+VERB+POS+PAST+A3SG

Figure 5.5: Erroneous morphological analyses for predicate prediction.

For the first problem, we correct wrong analyses for words detected in predicate list. On the other hand for multiple word predicates, we lost the annotation transfer chance for the sentence if the verb part is not aligned with the IBM Models. To solve this problem we tried to use shallow parse information while selecting predicates. Whenever we find a verb predicate, we added any word in its

constituent boundary to predicate candidates even the word is not in the desired morphological pattern.

We run our procedure with IBM Model 1 & 2 separately and present results in Table 5.5 and 5.6. In these runs, annotation transfer with IBM Model 2 gave slightly better output. There exists 14672 correct, 7300 incorrect annotations. 10809 words transferred but undetermined since hand annotated version do not have annotations for these words. However, 7202 words are not transferred while hand annotated corpus has valid annotation for the same words. Also, 4726 words are not transferred but there is not matching valid annotation in the hand annotated corpus. If we take correct, incorrect and untransferred words into account, annotation transfer with IBM Model 2 yields 50.29% true annotation. Results of the annotation transfer using IBM Model 1 are as follows; 14449 correct, 7223 incorrect annotations and also 10565 undetermined annotations. On the other hand 12472 words is not transferred at all. 4970 words are not transferred and there is no annotation in hand annotated corpus. If we take the same set into account, annotation transfer with IBM Model 1 yields ~50% true annotation. The number of predicates found by using model 1 is slightly more than the ones found with model 2. The number of predicates found with model 1 is 5191 and model 2 has 5133 correct predicate prediction, where total number of predicate in the phrase sentences are 6715. These counts are fairly better than WordNet alignment but they still need improvement.

Table 5.5: Results of annotation transfer using IBM Model 1 as alignment method.

| **IBM Model 1** | | | |
|---|---|---|---|
| **Transferred** | | **Untransferred** | |
| Correct | 14449 | Invalid HA | 4970 |
| Incorrect | 7223 | Valid HA | 7502 |
| Undetermined | 10565 | | |
| **Total** | 32237 | **Total** | 12472 |

To increase precision of the transfer, we add reinforce step previously used in Section 5.2.2. This increase the precision to 58.69% for model 1 and 59.18% for

Table 5.6: Results of annotation transfer using IBM Model 2 as alignment method.

| **IBM Model 2** | | | |
|---|---|---|---|
| **Transfered** | | **Untransfered** | |
| Correct | 14672 | Invalid HA | 4726 |
| Incorrect | 7300 | Valid HA | 7202 |
| Undetermined | 10809 | | |
| **Total** | 32781 | **Total** | 11928 |

model 2. Detailed results can be seen in Table 5.7 and 5.8. Although reinforce step increase correct annotations, it also introduce many new incorrect annotations. It also increase found predicate count by 49 for model 1 and by 54 for model 2 alignment.

Table 5.7: Results after reinforce step for IBM Model 1 alignment.

| **IBM Model 1 + Reinforce** | | | |
|---|---|---|---|
| **Transferred** | | **Untransferred** | |
| Correct | 17123 | Invalid HA | 1382 |
| Incorrect | 9505 | Valid HA | 2546 |
| Undetermined | 14153 | | |
| **Total** | 40781 | **Total** | 3928 |

Table 5.8: Results after reinforce step for IBM Model 2 alignment.

| **IBM Model 2 + Reinforce** | | | |
|---|---|---|---|
| **Transfered** | | **Untransfered** | |
| Correct | 17264 | Invalid HA | 1291 |
| Incorrect | 9471 | Valid HA | 2439 |
| Undetermined | 14244 | | |
| **Total** | 40979 | **Total** | 3730 |

After examining language structure we decided to add rules to tag any untagged words after annotation transfer. We observed argument types affect noun inflections, for some argument types the last word in constituent boundary is taking certain suffixes. So first we find untagged word and select the last word in its constituent boundary. Since we run reinforce step beforehand, only untagged constituents exists in the sentence. In this respect, we set the following rules to determine argument annotation for untransfered words;

- For nouns and proper nouns:
  - Have no suffix then ARG0
  - Last morpheme tag is “ACCUSATIVE” (-(y)H, -nH) or “DATIVE” (-(y)A, -nA) then ARG1
  - Last morpheme tag is “LOCATIVE” (-DA, -nDA) or “ABLATIVE” (-DAn, -nDAn ) then ARGMLOC
  - Last morpheme tag is “INSTRUMENTAL” (-(y)lA) then ARG2
- For all word types
  - Morphological parse contains date, time then ARGMTMP
  - Morphological parse contains cardinal number, fraction, percent, range, real number, ordinal number then ARGMEXT

We use these rules to tag any untagged word. After applying these rules annotation transfer result is as shown in Table 5.9 and 5.10. Results show that rules applied slightly change the correct annotations. For model 1 rules output much more correct annotation than the incorrect ones whereas in model 2 the number of correct and incorrect annotations gathered are nearly same. However, precision for model 1 is improved to 59.44% and for model 2 precision become 59.86%.

Table 5.9: Results after applying rules for IBM Model 1 alignment.

| **IBM Model 1 + Reinforce + Rules** | | | |
|---|---|---|---|
| **Transferred** | | **Untransferred** | |
| Correct | 17340 | Invalid HA | 1151 |
| Incorrect | 9664 | Valid HA | 2170 |
| Undetermined | 14384 | | |
| **Total** | 41388 | **Total** | 3321 |

Although we detect most of the predicate candidates, correct annotation count is still insufficient. We check predicate annotation using hand annotated corpus. Manually tagged corpus has 6715 predicates. 5240 predicates are annotated with model 1 and 5187 predicates with model 2. Remaining predicates are unavailable

Table 5.10: Results after applying rules for IBM Model 2 alignment.

| **IBM Model 2 + Reinforce + Rules** | | | |
|---|---|---|---|
| **Transfered** | | **Untransfered** | |
| Correct | 17464 | Invalid HA | 1078 |
| Incorrect | 9635 | Valid HA | 2075 |
| Undetermined | 14457 | | |
| **Total** | 41556 | **Total** | 3153 |

for annotation transfer. These numbers shows alignment with model 1 gives true mapping for 78% of the predicates and 77.24% correct alignment is derived with model 2.

Additionally, after the predicate alignment we need to get annotations for each predicate label and align English words to the words in Turkish phrase sentence. In the annotation transfer 33310 words that have argument annotations in English are processed for aligned predicates. 28855 words are aligned with a word in the target sentence using IBM Model 1. Also 367 of unaligned words are found by string or root match between sentences. As for annotation transfer with IBM Model 2, 28004 words aligned over 33310 and 435 words are annotated via string or root match. To sum up, annotation transfer for arguments is affected by the alignment procedure, but running reinforce method and applying rules afterwards softens this effect. The number of the words not found in the alignment is decreased to 3321 and 3153 after these methods. Also, there exist undetermined annotation transfers where we cannot adjudicate correctness since hand annotated corpus do not have a valid annotation. These annotations may affect correct annotation count substantially. Annotations transferred with alignment models can provide a basis for proposition bank creation in resource-scarce languages. Annotations may then be checked quickly by the annotators and proposition bank reach the final state.

# Chapter 6

# Future Work

During the preparation of this thesis, we have deeply surveyed the literature and find out different approaches such as PropBank studies for different languages, auto-generated PropBanks, SRL methods for the refinement of existing PropBanks and adapted our methods to lay the foundations of Turkish proposition bank. However, there is still much more work to do to improve the constructed proposition bank. Due to lack of time, the following approaches, methods and experiments are left for the future work.

First of all, more sentences from Penn Treebank II may be added to the analyzed 9560 sentences in this study. This will increase the size of Turkish PropBank corpus so that it will become more reliable and represent more syntactic and semantic variations about the language. However, this procedure involves multiple parties like translators for the translation of newly introduced sentences and annotators for both framing of new predicates and annotation of the sentences. Furthermore, manuel annotation is very time and resource consuming task, and processing of more sentences from Penn Treebank II is not addressed in this thesis.

Also, the constructed proposition bank is designed to have single annotation for each word as in the first version of English PropBank. Even though, we prepared label structure to be used with multiple annotations, we did not annotate words with respect to multiple predicates. Currently, argument labels we provided contains roles along with the synset id of the predicate. This infrastructure can be

used to encapsulate annotations for different predicate instances. Similarly latest version of English proposition bank added nominal and adjective predicates for the annotation procedure. Frame sets for these predicate types can be added to Turkish proposition bank and annotations with respect to these role sets can be performed. As in the latest version of English PropBank, frame sets for verbal, nominal and adjectival predicates can be unified to present semantic concepts.

In the construction and automatic generation of Turkish PropBank, we have extensively used the annotations in the English counterpart. However, we did not use frame files provided by English PropBank directly, as explained for the languages Dutch [15], French [17], German [18] [19] and Portuguese [25] in Section 2.3. This study may be interesting which may give information about the relation between target and source languages.

For most of the languages having proposition bank, semantic role labelers are trained with the corpus provided right after the proposition bank construction. Then sentences outside of the corpus are used for testing the validity of frame sets extracted for predicates. In this manner, system can be checked with diverse semantic and syntactic constructions. However, a wide ranging corpus is required to run such test and optimize the created proposition bank.

Moreover, we have presented PropBank studies for different languages where a complete explanation for the presented languages is missing in the literature. We are planning to publish information mentioned in Section 2.3. Likewise, comparison of Turkish proposition banks is not published in a journal which we will submit after the completion of the thesis.

# Conclusion

PropBank annotation is well-defined and applied to various different languages. We have surveyed the literature and deeply analyzed 15 proposition banks including English PropBank which is the foundation of the proposition bank studies. English PropBank was explained in detail, along with the Penn Treebank structure and proposition bank studies for the other languages throughout this study. As a morphologically rich language Turkish lacks such an extensive resource. Only preliminary work was done when we started this research. Our efforts can be counted as one of the first attempts for constructing the Turkish proposition bank. The main contribution of this thesis is the constructed Turkish PropBank [31] which contains 9560 sentences from the translated Penn Treebank II. We have used guidelines provided by the original PropBank and annotated only verbal predicates. Argument roles for 1914 verb senses were manually annotated using an in-house developed NLP Toolkit. Randomly selected 250 sentences from the corpus were used to ensure quality and reliability of the annotations. These 250 sentences were annotated by expert annotators and compared with the annotations from the constructed proposition bank. 1879 annotations among 2392 annotations were matched which corresponds to 79% agreement. Constructed proposition bank was accepted and published by the Turkish Journal of Electrical Engineering & Computer Sciences.

Another Turkish proposition bank [28], [29], [30] was recently published by Şahin and Adalı which contains annotation of 5635 sentences from ITU-METU-Sabancı Treebank (IMST) [74] and frame sets for 1285 verb senses. Although both proposition banks were constructed on different datasets, frame files were generated with the verb senses taken from the same resource (TDK). We have presented

the comparison and the matching process of the newly-created Turkish Proposition Banks in our research. We compared proposition banks by matching frame sets of the corresponding verb senses. Aforementioned our verb senses were identified by synset ids from WordNet. Therefore, verb senses of Şahin was matched with the Turkish WordNet first, then the synset ids found were used to map verb senses of two proposition bank. Turkish is a resource-scarce language in terms of linguistic resources, this mapping between two proposition bank gives the opportunity to improve both datasets. Matching frame files can be re-evaluated since they represent characteristics of different corpora. Different syntactic constructions may be found which results different argument roles for the verb senses. Also, frame files that do not exist in both study can be merged to create large-scale valency lexicon which represents much more semantic information for the language. Matching both propbanks with WordNet, enables transferring semantic information from different languages since WordNet contains mapping with English counterpart.

We conducted experiments on automatic proposition bank generation. First, we exploited the parallelism with English proposition bank. Aligned tokens in translated Penn Treebank II sentences were used for annotation projection. These sentences were in tree structure where English words share the same node with their Turkish translations. We enriched annotations of each tree node with the argument roles gathered from English proposition bank. Prior to annotation transfer we matched predicates in both English and Turkish sentences. A method was proposed for predicate prediction to detect Turkish predicates by looking morphological annotation of the predicate candidates. Words having verb root and verb according to last inflectional group was treated as predicates. After extracting predicate candidates, we checked the argument role of the word in English proposition bank. If the argument role is equal to "Predicate" in English annotation, we have transferred all annotations with respect to verb to the aligned Turkish words. The projected annotations then compared with the manually constructed

proposition bank. 65% of the roles are matched with the hand annotated corpus. This method can be used with any language with the parallel tree structure, however having such resource is difficult.

We also proposed methods for annotation projection to the parallel phrase sentences. Prior to the annotation transfer, tree structured English sentences were translated into Turkish without such structure. Translated sentences were also annotated manually for argument roles and shallow parse information by the annotators. In the first method, word pairs in both languages was aligned with WordNet's inter-lingual mapping. Again as the first step, we used the same predicate prediction method to detect predicate candidates in Turkish. Then, we tried to match English predicates to predicate candidates using semantic analysis of words in both sentence. English synset ids were mapped to Turkish verb sense using WordNet. After matching predicates, annotations in English sentence were transferred again using WordNet mapping for each node. However, due to missing semantic annotations in both corpus, mistranslations and missing mapping in WordNet, system only transferred a small set of the argument roles. Also, we used constituent boundaries to propagate transferred annotations. Untagged words were tagged with the maximum occurred argument role within its boundary. Correct annotations transferred after this method is ~14.59%. However, 4255 correct argument roles are transferred among 5457 arguments transferred which means 79% of the transferred roles are correct.

To increase annotation transfer for phrase sentences, we have also proposed alignment with IBM Model 1 and 2. The same sentences and predicate prediction method was used for the annotation projection. Both models yields ~50% correct annotations. Then, we have used constituent boundary information to tag unmatched words and integrated rules to determine argument roles which increases correct annotations to ~60%.

Automatic proposition bank generation methods proposed here can be improved by checking annotations for passive, unergative, and unaccusative verbs since

semantic interpretation of such verbs are different. Another issue affects annotation transfer is the erroneous annotations in different levels of the corpus. From translation to morphological analysis and sense selection any error is causing an avalanche effect and prevent transferring true annotations. Also, comparison of annotations in automatic proposition banks with the hand annotated corpus is not accurate for the roles Arg2-Arg5 since the semantic information concealed in these roles may not be identical.

While this study constituted a base for Turkish proposition bank, there are so many unanswered questions and work to do as we mentioned in Future Work chapter.

# References


[1] B. Levin. *English verb classes and alternations : a preliminary investigation*. of Chicago Press, University, 1993.

[2] K. Kipper, H. T. Dang, and M. Palmer. Class-based construction of a verb lexicon. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 691–696. AAAI Press, 2000. ISBN 0-262-51112-6.

[3] C. J. Fillmore, J. Ruppenhofer, and F. Baker, Collin. *FrameNet and Representing the Link between Semantic and Syntactic Relations*, pages 19–62. Language and Linguistics Monographs Series B. Institute of Linguistics, Academia Sinica, Taipei, 2004.

[4] P. Kingsbury and M. Palmer. From treebank to propbank. In *LREC*. European Language Resources Association, 2002.

[5] P. Kingsbury and M. Palmer. Propbank: The next level of treebank. In *Proceedings of Treebanks and Lexical Theories*, Vxj, Sweden, 2003.

[6] M. Palmer, D. Gildea, and P. Kingsbury. The proposition bank: An annotated corpus of semantic roles. *Comput. Linguist.*, 31(1):71–106, March 2005. ISSN 0891-2017.

[7] C. Bonial, J. Bonn, K. Conger, J. D. Hwang, and M. Palmer. Propbank: Semantics of new predicate types. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*. European Language Resources Association (ELRA), 2014.

[8] Meyers A., R. Reeves, C. Macleod, R. Szekely, V. Zielinska, B. Young, and R. Grishman. The nombank project: An interim report. In *HLT-NAACL 2004 Workshop: Frontiers in Corpus Annotation*, pages 24–31, Boston, Massachusetts, USA, May 2 - May 7 2004. Association for Computational Linguistics.

[9] M. Palmer, O. Babko-Malaya, A. Bies, M. Diab, M. Maamouri, A. Mansouri, and W. Zaghouani. A pilot arabic propbank. In *The Sixth International Language Resources and Evaluation Conference (LREC2008)*, May 2008.

[10] W. Zaghouani, M. Diab, A. Mansouri, S. Pradhan, and M. Palmer. The revised arabic propbank. In *Proceedings of the Fourth Linguistic Annotation Workshop*, LAW IV '10, pages 222–226, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. ISBN 978-1-932432-72-5.

[11] E. E. Agirre, I. Aldezabal, J. Etxeberria, and E. Pociello. A Preliminary Study for Building the Basque PropBank. In *Proceedings of the 5th International Conference on Language Resources and Evaluations (LREC)*, April 2006.

[12] I. Aldezabal, M. J. Aranzabe, A. Díaz de Ilarraza, and A. Estarrona. Building the basque propbank. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may 2010. European Language Resources Association (ELRA). ISBN 2-9517408-6-7.

[13] I. Aldezabal, M. J. Aranzabe, A. Díaz de Ilarraza, A. Estarrona, and L. Uria. Euspropbank: Integrating semantic information in the basque dependency treebank. In *Computational Linguistics and Intelligent Text Processing*, pages 60–73, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-12116-6.

[14] Nianwen X. and M. Palmer. Adding semantic roles to the chinese treebank. *Natural Language Engineering*, 15(1):143–172, 2009.

[15] O. De Clercq, V. Hoste, and P. Monachesi. Evaluating automatic cross-domain dutch semantic role annotation. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012*, pages 88–93, 2012.

[16] K. Haverinen, J. Kanerva, S. Kohonen, A. Missilä, S. Ojala, T. Viljanen, V. Laippala, and F. Ginter. The finnish proposition bank. *Language Resources and Evaluation*, 49(4):907–926, Dec 2015.

[17] L. van der Plas, T. Samardžić, and P. Merlo. Cross-lingual validity of propbank in the manual annotation of french. In *Proceedings of the Fourth Linguistic Annotation Workshop*, LAW IV '10, pages 113–117, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. ISBN 978-1-932432-72-5.

[18] K. Erk, A. Kowalski, S. Padó, and M. Pinkal. Towards a resource for lexical semantics: A large german corpus with extensive semantic annotation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 537–544, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[19] A. Burchardt, A. Frank, S. Pado, and M. Pinkal. The salsa corpus: a german corpus resource for lexical semantics. In *Proceedings of LREC 2006 : the 5 th International Conference on Language Resources and Evaluation, Genoa, Italy. - Paris*, pages 969–974, 2006.

[20] A. Vaidya, J. D. Choi, M. Palmer, and B. Narasimhan. Analysis of the hindi proposition bank using dependency structure. In *Proceedings of the 5th Linguistic Annotation Workshop*, LAW V '11, pages 21–29, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-932432-93-0.

[21] D. Kawahara, S. Kurohashi, and K. Hasida. Construction of a japanese relevance-tagged corpus. In *LREC*. European Language Resources Association, 2002.

[22] M. Palmer, S. Ryu, J. Choi, S. Yoon, and Y. Jeon. Korean propbank. Philadelphia: Linguistic Data Consortium, 2006.

[23] H. Song, C. Park, J. Lee, M. Lee, Y. Lee, J. Kim, and Y. Kim. Construction of korean semantic annotated corpus. In *Computer Applications for Database, Education, and Ubiquitous Computing - International Conferences, EL, DTA and UNESST 2012, Held as Part of the Future Generation Information Technology Conference, (FGIT) 2012, Gangneug, Korea, December 16-19, 2012. Proceedings*, pages 265–271, 2012.

[24] A. Mirzaei and A. Moloodi. Persian proposition bank. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016.*, 2016.

[25] M. S. Duran and S. M. Aluísio. Propbank-br: a brazilian treebank annotated with semantic role labels. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may 2012. European Language Resources Association (ELRA). ISBN 978-2-9517408-7-7.

[26] L. Màrquez, L. Villarejo, M. A. Martí, and M. Taulé. Semeval-2007 task 09: Multilevel semantic annotation of catalan and spanish. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval '07, pages 42–47, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics.

[27] Taulé M., M. A. Martí, and M. Recasens. Ancora: Multilevel annotated corpora for catalan and spanish. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC-08)*, Marrakech, Morocco, May 2008. European Language Resources Association (ELRA).

[28] G. G. Şahin. Framing of verbs for turkish propbank. In *TurCLing 2016 in conj. with 17th International Conference on Intelligent Text Processing and Computational Linguistics (CICLING 2016)*, 2016.

[29] G. G. Şahin. Verb sense annotation for turkish propbank via crowdsourcing. In *17th International Conference on Intelligent Text Processing and Computational Linguistics (CICLING 2016)*, 2016.

[30] G. G. Şahin and E. Adalı. Annotation of semantic roles for the Turkish proposition bank. *Language Resources and Evaluation*, May 2017. ISSN 1574-0218.

[31] K. Ak, O. T. Yıldız, V. Esgel, and C. Toprak. Construction of a Turkish proposition bank. *Turkish Journal of Electrical Engineering and Computer Science*, 26:570 – 581, 2018. ISSN 1300-0632.

[32] S. Mukund, D. Ghosh, and R. K. Srihari. Using cross-lingual projections to generate semantic role labeled corpus for urdu: A resource poor language. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 797–805, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[33] K. Ak, Ö. Bakay, and O. T. Yıldız. Comparison of turkish proposition banks by frame matching. In *International Conference on Computer Science and Engineering (UBMK)*, 09 2018.

[34] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330, 1993.

[35] O. Babko-Malaya. *Guidelines for Propbank Framers*, 2005. URL `http://verbs.colorado.edu/~mpalmer/projects/ace/FramingGuidelines.pdf`.

[36] M. Haspelmath. More on the typology of inchoative/causative verb alternations. In *Causatives and transitivity*, Studies in Language Companion

Series, 23, pages 87–120. John Benjamins Publishing Company, Amsterdam, Netherlands, 01 1993.

[37] O. Babko-Malaya. *Propbank Annotation Guidelines*, 2005. URL `http://verbs.colorado.edu/~mpalmer/projects/ace/PBguidelines.pdf`.

[38] P. Martha. SemLink: Linking PropBank, VerbNet and FrameNet. In *Proceedings of the Generative Lexicon ConferenceGenLex-09*, Pisa, Italy, Sept 2009.

[39] M. Maamouri and A. Bies. Developing an arabic treebank: Methods, guidelines, procedures, and tools. In *Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages*, Semitic '04, pages 2–9, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[40] T. Buckwalter. Arabic morphological analyzer version 1.0. Linguistic Data Consortium, 2002.

[41] J. D. Choi, C. Bonial, and M. Palmer. Propbank frameset annotation guidelines using a dedicated editor, cornerstone. In *LREC*, 2010.

[42] J. D. Choi, C. Bonial, and M. Palmer. Propbank instance annotation guidelines using a dedicated editor, jubilee. In *LREC*, 2010.

[43] I. Aldezabal. *Aditz-azpikategorizazioaren azterketa. 100 aditzen azterketa zehatza, Levin (1993) oinarri harturik eta metodo automatikoak baliatuz. Doktorego txostena.* PhD thesis, Euskal Filologia saila., Leioa, 2004.

[44] I. Aduriz, M. J. Aranzabe, J. M. Arriola, A. Atutxa, A. Díaz de Ilarraza, A. Garmendia, and M. Oronoz. Construction of a basque dependency treebank. In *In Joakim Nivre and Erhards Hinrichs eds., Proceedings of the Second Workshop on Treebanks and Linguistic Theories, pp.: 201-204, ISSN: 1651-0267, ISBN: 91-7636-394-5, TLT 2003, Vaxjo, Sweden, November 14-15.*, 2003.

[45] A. Díazde Ilarraza Sánchez, A. Garmendia, and M. Oronoz. Abar-hitz: An annotation tool for the basque dependency treebank. In *LREC*. European Language Resources Association, 2004.

[46] I. Aldezabal. *Levin's classes and Basque. A comparative approach.* PhD thesis, University of Maryland., UMIACS. Departamental colloquia., 1998.

[47] N. Xue. A chinese semantic lexicon of senses and roles. *Language Resources and Evaluation*, 40(3):395–403, Dec 2006. ISSN 1572-0218.

[48] N. Xue, F. Xia, F. Chiou, and M. Palmer. The penn chinese treebank: Phrase structure annotation of a large corpus. *Nat. Lang. Eng.*, 11(2):207–238, June 2005. ISSN 1351-3249.

[49] M. Reynaert, N. Oostdijk, O. De Clercq, H. van den Heuvel, and F. M.G. de Jong. Balancing sonar: Ipr versus processing issues in a 500-million-word written dutch reference corpus. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, pages 2693–2698. European Language Resources Association (ELRA), 5 2010.

[50] P. Pajas and J. Štěpánek. Recent advances in a feature-rich framework for treebank annotation. In *Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pages 673–680, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics. ISBN 978-1-905593-44-6.

[51] K. Haverinen, J. Nyblom, T. Viljanen, V. Laippala, S. Kohonen, A. Missilä, S. Ojala, T. Salakoski, and F. Ginter. Building the essential resources for finnish: the turku dependency treebank. *Language Resources and Evaluation*, 48(3):493–531, Sep 2014. ISSN 1574-0218.

[52] S. Brants, S. Dipper, S. Hansen, W. Lezius, and G. Smith. The TIGER treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*, Sozopol, 2002.

[53] R. Bhatt, B. Narasimhan, M. Palmer, O. Rambow, D. M. Sharma, and F. Xia. A multi-representational and multi-layered treebank for hindi/urdu. In *Proceedings of the Third Linguistic Annotation Workshop*, ACL-IJCNLP '09, pages 186–189, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[54] M. Civit and M. A. Martí. Building cast3lb: A spanish treebank. *Research on Language and Computation*, 2(4):549–574, Dec 2004. ISSN 1572-8706.

[55] M. A. Martí and M. Taulé. Cess-ece treebanks. *Publicacions de la Universitat de Barcelona.*, 2008.

[56] M. S. Rasooli, M. Kouhestani, and A. Moloodi. Development of a persian syntactic dependency treebank. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 306–314. Association for Computational Linguistics, 2013.

[57] S. Afonso, E. Bick, R. Haber, and D. Santos. Floresta sintá(c)tica: A treebank for portuguese. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*. European Language Resources Association (ELRA), 2002.

[58] A. Burchardt, K. Erk, A. Frank, A. Kowalski, S. Padó, and M. Pinkal. Salto–a versatile multi-level annotation tool. In *Proceedings of LREC*, volume 2006, pages 517–520. Citeseer, 2006.

[59] A. Branco, C. Carvalheiro, S. Pereira, S. Silveira, J. Silva, S. Castro, and J. Graça. A propbank for portuguese: the cintil-propbank. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may 2012. European Language Resources Association (ELRA). ISBN 978-2-9517408-7-7.

[60] J. Silva, A. Branco, S. Castro, and R. Reis. Out-of-the-box robust parsing of portuguese. In *Computational Processing of the Portuguese Language*,

pages 75–85, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-12320-7.

[61] S. Oepen. [incr tsdb()] — competence and performance laboratory. User manual. Technical report, Computational Linguistics, Saarland University, Saarbrücken, Germany, 2001. in preparation.

[62] M. Anwar, R. A. Bhat, D. Sharma, A. Vaidya, M. Palmer, and T. A. Khan. A proposition bank of urdu. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, may 2016. European Language Resources Association (ELRA). ISBN 978-2-9517408-9-1.

[63] R. A. Bhat and D. M. Sharma. A dependency treebank of urdu and its evaluation. In *Proceedings of the Sixth Linguistic Annotation Workshop*, LAW VI '12, pages 157–165, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[64] R. A. Bhat, N. Jain, A. Vaidya, M. Palmer, T. A. Khan, D. M. Sharma, and J. Babani. Adapting predicate frames for urdu propbanking. In *Proceedings of the EMNLP'2014 Workshop on Language Technology for Closely Related Languages and Language Variants*, pages 47–55. Association for Computational Linguistics, 2014.

[65] T. Zhuang and C. Zong. Joint inference for bilingual semantic role labeling. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 304–314, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[66] L. van der Plas, P. Merlo, and J. Henderson. Scaling up automatic cross-lingual semantic role annotation. In *ACL (Short Papers)*, pages 299–304. The Association for Computer Linguistics, 2011.

[67] M. Kozhevnikov and I. Titov. Cross-lingual transfer of semantic role labeling models. In *ACL (1)*, pages 1190–1200. The Association for Computer Linguistics, 2013.

[68] L. van der Plas, M. Apidianaki, and C. Chen. Global methods for cross-lingual semantic role and predicate labelling. In *COLING*, pages 1279–1290. ACL, 2014.

[69] M. R. Gormley, M. Mitchell, B. Van Durme, and M. Dredze. Low-resource semantic role labeling. In *ACL (1)*, pages 1177–1187. The Association for Computer Linguistics, 2014.

[70] A. Akbik, L. Chiticariu, M. Danilevsky, Y. Li, S. Vaithyanathan, and H. Zhu. Generating high quality proposition banks for multilingual semantic role labeling. In *ACL (1)*, pages 397–407. The Association for Computer Linguistics, 2015.

[71] O. T. Yıldız, E. Solak, O. Görgün, and R. Ehsani. Constructing a turkish-english parallel treebank. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 2, pages 112–117, 2014.

[72] O. T. Yıldız, E. Solak, Ş. Çandır, R. Ehsani, and O. Görgün. Constructing a turkish constituency parse treebank. In *Information Sciences and Systems 2015*, pages 339–347. Springer, 2015.

[73] C. Acartürk and D. Zeyrek. Unaccusative/unergative distinction in turkish: A connectionist approach. In *Proceedings of the Eighth Workshop on Asian Language Resouces*, pages 111–119, 2010.

[74] U. Sulubacak, T. Pamay, and G. Eryiğit. Imst: A revisited Turkish dependency treebank. In *The First International Conference on Turkic Computational Linguistics*, pages 1–6, 2016.

[75] J. D. Choi, C. Bonial, and M. Palmer. Propbank frameset annotation guidelines using a dedicated editor, cornerstone. In *LREC*, 2010.

[76] R. Ehsani, E. Solak, and O. T. Yıldız. Constructing a wordnet for Turkish using manual and automatic annotation. *ACM Transactions no Asian Low-Resource Language Information Processing*, 17(3), 2018.

[77] M. Palmer, P. Kingsbury, O. Babko-Malaya, S. Cotton, and B. Snyder. Proposition bank i. Philadelphia: Linguistic Data Consortium, 2004. LDC2004T14.

[78] P. F. Brown, V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Comput. Linguist.*, 19(2):263–311, June 1993. ISSN 0891-2017.

# Appendix A

# Diagnostics for Unergative/Unaccusative Distinction in Turkish

Intransitives is generally accepted to have two subclasses unergatives and unaccusatives. The distinction between these classes is achieved in numerous ways. In some languages perfective auxiliaries are used for this task. However, they are absent for Turkish language. Some diagnostics proposed in [73] for distinction. These methods are not strict as auxiliaries in other languages, and some of the verbs may not be classified correctly. (Examples are taken from [73]).

## A.1 The *"-ArAk"* Construction

In Turkish, verbs can take "-ArAk" suffix and constitute adverbial clause. In Turkish, if a sentence contains a verb with this construction, predicate and verb that has the suffix should be in the same class. Which means they are both unergative or unaccusative, combination of two classes is not valid. In Figure A.1, both of the verbs in 13 unaccusative where in 14 verbs are unergative. Last two examples are ungrammatical where verbs unergative-unaccusative 15 and unaccusative-unergative 16 verb pairs used.

(13) Hasan [kol-u kana -y -arak] acı çek-ti.
arm-POSS bleed-GL-ArAk suffer-PST
"Hasan, while his arm bled, suffered."

(14) Kız [ (top) oyna-y -arak] şarkı söyledi.
girl ball play-GL-ArAk sing-PST
"The girl, while playing (ball), sang."

(15) * Kız [ (top) oyna-y -arak] kay-dı.
girl ball play-GL-ArAk slip -PST
"The girl, while playing (ball), slipped."

(16) * Kız [kayak kay-arak] düş-tü.
girl ski-ArAk fall-PST
"The girl, while skiing, fell."

Figure A.1: Verbs with *"-ArAk"* construction occur with the predicates of the same class.

## A.2 Double Causatives

In Turkish, verbs can take causative suffixes like "-Ar", "-DHr", "-Hr", "-Ht", "-t" and change meaning to have something done by somebody or to get something done by something. One of the diagnostics for Split Intransitivity for Turkish is only unaccusative can take double causative suffixes, unergatives can not. In Figure A.2, soldurttu in 17 is unaccusative and take double causative suffixes, where in 18 double construction for unergative verb is ungrammatical.

(17) Sema Turhan-a çiçeğ-i sol- dur -t -tu.
-DAT flower-ACC fade-CAUS-CAUS-PST
"Sema made Turhan cause the flower to fade."

(18) * Ben Turhan-a Sema-yı koş-tur -t -t -um
I -DAT -ACC run-CAUSCAUS-PST-1sg
"I made Turhan make Sema run."

Figure A.2: Double causative constructions for unaccusative and unergative verbs.

## A.3 Gerund Constructions

Some gerund constructions is specific to verb class. Suffix "-Irken", used instead of "while", is used for unergative verbs where "-Ince" stands for "when" is applicable for unaccusative verbs. In Figure A.3, "-Irken" suffix is used in 19 for unergative verb çalışmak and "-Ince" is added to the unaccusative verb takılmak in 20.

(19) Adam çalış-ırken esne-di.
man work-Irken yawn-PAST.3per.sg
"The man yawned while working."

(20) Atlet takıl-ınca düş-tü.
athlete trip-IncE fall-PAST.3per.sg
"The athlete when tripped fell."

Figure A.3: Gerund constructions for unaccusative and unergative verbs.

## A.4 The Suffix *"Ik"*

Another diagnostic for SI is suffix "-Ik" which derives adjectives from verbs. It is only applicable to unaccusative verbs and ungrammatical for unergatives. In 22, usage is ungrammatical since "-Ik" is attached to an unergative verb.

(21) bat-ık gemi
sink-Ik ship
the sunk ship

(22) *çalış-ık adam
work-Ik man
the worked man

Figure A.4: *"Ik"* suffix only compatible with unaccusatives.

## A.5 The *"mIş"* Participle

Yet another suffix deriving adjective from verbs is the past participle marker "-mIş" suffix. It is compatible with transitive, intransitive and also passivized

verbs. For intransitive verbs the "mIş" suffix is more prevalent than unergatives. In example 23 unaccusative verb with "mIş" contruction is shown. Usage of the example verbs in 23 is more favorable than the unergative verbs in 24.

(23) sol-muş/ karar-mış çiçek
wilt/ blacken -mIş flower
"The wilted/blackened flower"

(24) *sıçra-mış/ yüz-müş/ bağır-mış çocuk
jump/ swim/ shout -mIş child
"The jumped/ swum/ shouted child"

Figure A.5: *"mIş"* participle with unaccusative and unergative verbs.

## A.6 Impersonal Passivization

In different languages impersonal passivization is used for split intransitivity. It can also be used for distinguishing unergative verbs for Turkish too. The passive suffix marker in Turkish is "-Il" which is generaly exists with agentive by-phrase. The acceptance of impersonal passives is affected by the tense of the verb. A generic or existential interpretation is assigned to the implicit subject of the verbs in aorist form.

(25) Burada koşuldu.
Here run-PASS-PST
"There was running here." (existential interpretation)

(26) ??Bu yetimhanede büyündü.
This orphanage-LOC grow-PASS-PST
"It was grown in this orphanage."

(27) *Burada ölündü.
here die-PASS-PST
"Here people die."

Figure A.6: Impersonal passivization of unaccusative and unergative verbs.

Hovewer for the verbs in past tense, implicit subject is in first person plural reading. Therefore verbs in past tense is distinguishable with impersonal passivization. Unergative verbs as in 25 can undergo impersonal passivization where examples 26 and 27 are not grammatical.

Diagnostics listed here may not hold for every verb. Some of the verbs with human subject like “düş”, “gel”, “gir” is unaccusative with most of the previous diagnostics but unergative for impersonal passivization. Same thing also exists for the opposite, where verb “devam et” is unergative with most of the diagnostics but unaccusative for impersonal passivization.

# Curriculum Vitae