ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

BUILDING OF TURKISH PROPBANK AND SEMANTIC ROLE LABELING OF TURKISH

Ph.D. THESIS

Gözde Gül ŞAHİN

Faculty of Computer and Informatics Engineering

Computer Engineering Programme

FEBRUARY 2018

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

BUILDING OF TURKISH PROPBANK AND SEMANTIC ROLE LABELING OF TURKISH

Ph.D. THESIS

Gözde Gül ŞAHİN
(504122519)

Faculty of Computer and Informatics Engineering

Computer Engineering Programme

Thesis Advisor: Prof. Dr. Eşref ADALI

FEBRUARY 2018

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

TÜRKÇE ÖNERME VERİ TABANININ OLUŞTURULMASI VE TÜRKÇENİN GÖREV ÇÖZÜMLEMESİ

DOKTORA TEZİ

Gözde Gül ŞAHİN
(504122519)

Bilgisayar ve Bilişim Fakültesi

Bilgisayar Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Eşref ADALI

ŞUBAT 2018

Gözde Gül ŞAHİN, a Ph.D. student of ITU Graduate School of Science Engineering and Technology 504122519 successfully defended the thesis entitled "BUILDING OF TURKISH PROPBANK AND SEMANTIC ROLE LABELING OF TURKISH", which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Prof. Dr. Eşref ADALI** Istanbul Technical University	.....
Jury Members :	**Prof. Dr. Tunga Güngör** Boğaziçi University	.....
	Doç. Dr. Deniz Yüret Koç University	.....
	Doç. Dr. A. Cüneyd Tantuğ Istanbul Technical University	.....
	Yrd. Doç Dr. Gülşen Eryiğit Istanbul Technical University	.....

Date of Submission : **16 February 2018**
Date of Defense : **16 January 2018**

To my family and friends,

FOREWORD

I consider myself so very lucky for meeting my advisor, Eşref Adalı during my undergraduate years. He has been a guide, life coach, teacher, friend, fellow traveller, role model and many more things to me. I am glad our paths crossed and you have convinced me to undertake a research career. I am extremely grateful to you for introducing me to this exciting research area that changed my life forever.

It has been a long journey with many ups and downs. In the most hopeless times, Gülşen Eryiğit always had a good advice or an oppurtunity that would keep me going. Thank you for teaching me a great deal on NLP, statistics; thank you for your openness, inspiration, hard work and thank you for being such a strong role model for future women researchers. This thesis would not be possible without your support.

I am sincerely grateful to Mark Steedman who made my visit to University of Edinburgh possible. Your wisdom, enthusiasm and knowledge in wide range of areas have amazed and inspired me supremely. Thank you for your guidance, support and making me feel welcome. I would like to thank Adam Lopez, who has been a great mentor and support. Thanks for changing my way of thinking about research, triggering fruitful discussions, including me in your group meetings and offering me help. Andrew, Ben, Carmen residents of 3.32: Clara, Irene, Mona, Sameer and Spandana; "steedman-students" and members of ILCC: I learned a lot from each of you. Thanks for your friendship, support, discussions and making my visit unforgettable.

I would like to thank to all members of İTÜ Natural Language Processing Group and the faculty members of İTÜ Computer and Informatics Engineering for providing me with necessary tools and knowledge for my research and a warm, collaborative environment. My special thanks go to my friends, colleagues and labmates: Gönül, Mahiye, İlknur, Ahmet, Müge, İsmail, Ezgi, Payam, Dilara, Kübra, Tuğba, Memduh, Umut, Resul, Hakan, Emrullah for making graduate life bearable by turning the department into a fun and warm place.

My special thanks go to my colleague and my dear friend Fatih from PragmaCraft for brainstorming, paper discussions and coding sessions. It has been a great pleasure to work and learn together with you.

I would like to thank Cüneyd Tantuğ, Tunga Güngör for being members of the thesis follow-up committee and defense jury. Their precious feedbacks contributed to the work in this dissertation.

I would like to express my gratitude to the Scientific and Technical Research Council of Turkey (TÜBİTAK), who has supported my PhD studies and my visit to University of Edinburgh through TUBITAK-2211 Domestic PhD Scholarship and TUBITAK-2214/A Graduate Scholarship for Outgoing Students Programs. I would like to thank Linguistic Data Consortium (LDC) for providing me with CoNLL-2009 shared task corpora free of charge via their Data Scholarship Program. I owe special thanks to builders of IMST for providing me an early access; TNC (Turkish National

Corpus) for the query interface account and to Belgin Aksu from TDK for sharing a valuable dataset on Turkish verb-argument structure.

Finally, thanks to my dearest friends Cansu and Selcen for their patience and emotional support; my family for believing in me more than I believe in myself and my husband, Celal, for his love, support, encouragment, patience, understanding and trust in me.

February 2018

Gözde Gül ŞAHİN
Yük. Müh.

TABLE OF CONTENTS

Page

FOREWORD ..... **x**
TABLE OF CONTENTS ..... **xi**
ABBREVIATIONS ..... **xv**
LIST OF TABLES ..... **xvii**
LIST OF FIGURES ..... **xxi**
SUMMARY ..... **xxiii**
ÖZET ..... **xxv**
1. Introduction **1**
1.1 Statement of the Problem 5
1.2 Main Contributions of the Thesis 6
1.3 Organization of the Thesis 7
2. Framing of Turkish **11**
2.1 Background and Related Work 11
2.2 Method 12
2.2.1 Framing Tool 14
2.2.2 Distinguishing Senses 14
2.2.2.1 Light Verbs and Multiword Expressions (MWE) 16
2.2.3 Semantic Role Numbering 17
2.3 Morphosemantics 18
2.3.1 Case Marking 19
2.3.2 Derivational Morphology 20
2.3.2.1 Valency Changes 24
2.3.2.2 Nominal Verbs 27
2.4 Lexicon Statistics 28
2.5 Summary 28
3. Annotation of Turkish **29**
3.1 Background and Related Work 29
3.2 Corpus 30
3.3 Feasibility 32
3.3.1 Verbal Nominals 32
3.3.2 Nominal Verbs 34
3.3.3 Copula 35
3.4 Crowdsourcing Linguistic Annotation 35
3.4.1 Verb Sense Disambiguation (VSD) Task 36
3.4.1.1 VSD Input Design for CrowdFlower 37
3.4.2 Semantic Role Annotation (SRA) Task 38
3.4.3 Configuration 41

3.4.4 Quality Control ... 41
3.5 Results ... 43
3.5.1 VSD Results ... 43
3.5.2 SRA Results ... 45
3.6 Post Annotation ... 51
3.7 Discussion ... 52
3.8 Dataset Statistics ... 55
3.9 Summary ... 57
4. Statistical Turkish Semantic Role Labeling ... 59
4.1 Background and Related Work ... 59
4.1.1 Data set and Evaluation ... 60
4.1.2 Logistic Regression (LR) ... 61
4.1.3 Reranking ... 62
4.2 Method ... 62
4.2.1 Features ... 64
4.3 Research Questions ... 66
4.4 Experiments ... 67
4.4.1 Q1: How important is morphosemantic features? ... 67
4.4.2 Q2: How much training data is required? ... 68
4.4.3 Q3: Are high-level features actually needed for SRL ? ... 71
4.4.4 Q4-Q5: Contribution of Continuous Features ... 71
4.5 Analysis of Experiments ... 72
4.5.1 PD Analysis ... 73
4.5.2 Argument Labeling Analysis ... 74
4.6 Summary ... 75
5. Neural Turkish Semantic Role Labeling ... 79
5.1 Background and Related Work ... 80
5.2 Method ... 83
5.2.1 Subword Units ... 87
5.2.2 Composition Methods ... 88
5.2.3 Multiple Subword Units ... 90
5.2.3.1 Apriori Integration ... 90
5.2.3.2 Post Integration ... 91
5.3 Experiments ... 92
5.3.1 Single Unit: Turkish ... 93
5.3.2 Single Unit: German, Spanish, Czech, Catalan, Finnish ... 95
5.3.3 Multiple Units: Turkish ... 96
5.3.4 Multiple Units: German, Spanish, Czech, Catalan, Finnish ... 98
5.4 Analysis of Experiments ... 99
5.4.1 Label Error Analysis ... 100
5.4.2 Comparison with statistical model ... 102
5.4.2.1 Weaknesses and Strengths ... 103
5.4.3 Dataset Experiment ... 104
5.5 Summary ... 104
6. Conclusion and Future Work ... 107

REFERENCES **109**
APPENDICES **119**
APPENDIX A.1 121
APPENDIX A.2 122
APPENDIX A.3 123
APPENDIX A.4 124
APPENDIX A.5 125
APPENDIX A.6 126
APPENDIX A.7 127
APPENDIX A.8 128
APPENDIX A.9 129
CURRICULUM VITAE **133**

ABBREVIATIONS

ABL : Ablative
AC : Argument Classification
ACC : Accusative
AI : Argument Identification
AM : Argument Modifier
AMR : Abstract Meaning Representation
BPE : Byte Pair Encoding
CNN : Convolutional Neural Network
CoNLL : Conference on Natural Language Learning
DAT : Dative
DB : Derivational Boundary
DNN : Deep Neural Network
FN : FrameNet
FOL : First Order Logic
IG : Inflectional Group
IMST : İTÜ-Metu-Sabancı Treebank
LM : Language Modeling
LOC : Locative
LSTM : Long-Short Term Memory
LV : Light Verb
MST : Metu-Sabancı Treebank
MT : Machine Translation
MWE : Multi Word Expression
NN : Neural Network
NOM : Nominative
OOV : Out of Vocabulary
PB : PropBank
PD : Predicate Disambiguation
PSD : Predicate Sense Disambiguation
PI : Predicate Identificaiton
POS : Parts of Speech
RNN : Recurrent Neural Network
SG : Stack Generalization
SGD : Stochastic Gradient Descent
SRA : Semantic Role Annotation
SRL : Semantic Role Labeling
ST : Shared Task
TDK : Türk Dil Kurumu (Turkish Language Association)
UD : Universal Dependencies
VSD : Verb Sense Disambiguation
VN : VerbNet

LIST OF TABLES

Page

Table 2.1 : Example framesets of “çalış” 13
Table 2.2 : Excluded root verbs and their frequencies in a million..... 15
Table 2.3 : Framing of the verb “ver- (to give)”..... 17
Table 2.4 : Thematic roles commonly associated with numbered arguments..... 18
Table 2.5 : The complete list of semantic labels for temporary roles 18
Table 2.6 : Case marking across languages (taken from World Atlas of Language Structures) 19
Table 2.7 : Case marking in Turkish and Hungarian..... 19
Table 2.8 : Relation between case markers and semantic roles. Ag: agent, Th: theme, Dest: destination, Sou: source, Pat: Patient 20
Table 2.9 : Relation between case markers and word senses 21
Table 2.10 : Derivational morphology of verbs in IMST. *Count*: Number of occurences in the treebank; ROOT: Root Verb; PASS: Passive; CAUS: Causative; RECIP: Reciprocal; *Adj*: Adjective; *Adv*: Adverb; Morphemes used in examples: *Neces*: Necessity; *Fut*: Future Tense; *Acc*: Accusative marker ;*Neg*: Negation; *Inf*: Infinitive 23
Table 2.11 : Framesets for mix.01 and karış.01 25
Table 2.12 : Example sentences for reciprocal verb *döv-üş (fight)* with (a) plural agent and (b) co-agent..... 26
Table 2.13 : #sense: Number of senses; #lemma: Number of lemmas..... 28
Table 2.14 : Columns: number of roles, Rows: number of senses..... 28
Table 3.1 : Input row for Crowdsourcing Task..... 37
Table 3.2 : Configuration of Verb Sense Annotation Task..... 41
Table 3.3 : Precision, Recall and F1 scores of crowd annotations. *Test*: Expert labeled 10%, *Other*: Unlabeled 90%, *All*: Combined 100% 46
Table 3.4 : Fleiss κ for each Semantic Role Category. # denotes number of occurrences 47
Table 3.5 : Confusion matrix for argument labels..... 49
Table 3.6 : Confusion matrix for secondary tags. Entries are a fraction of the total number of arguments, including core arguments..... 50
Table 3.7 : Semantic layer statistics on IMST. #distLemma: number of distinct predicate lemmas, #distSense: number of distinct predicate senses..... 55
Table 3.8 : Argument counts in Turkish PropBank..... 56
Table 4.1 : CoNLL 09 tabular format for SRL..... 61
Table 4.2 : Continuous features based on pretrained word representations proposed by Roth and Lapata [87]..... 65
Table 4.3 : Discrete Feature List 66

Table 4.4 : Labeled F1 scores for baseline Turkish SRL system. +:addition of feature to previous system; -Word+Lemma:substitution of word features with lemma features. F1: Score of original Propbank; F1-UD: Score of UD compliant PropBank 68
Table 4.5 : Effects of information level of the features 71
Table 4.6 : Effect of the continuous features. AL: Argument Labeling (AI+AC), OA: Overall system (PD+AI+AC), Org: Best system with only discrete features. 72
Table 4.7 : PD Errors vs Derivation Types 73
Table 4.8 : Argument Labeling performance per category 75
Table 5.1 : w values of Eq. 5.1 calculated for different languages (Universal Dependency Treebanks (UDT) are used for calculation). 81
Table 5.2 : ρ functions and outputs 88
Table 5.3 : PD accuracy results for for different subword units composed with addition, bi-LSTM and add-bi-LSTM. 93
Table 5.4 : Joint PI+PD results for different subword units composed with addition, bi-LSTM and add-bi-LSTM. 94
Table 5.5 : Labeled argument labeling scores for different subunits and composition functions, Best F1 for each composition is shown in bold. 94
Table 5.6 : Unlabeled argument labeling scores for different subunits and composition functions. Best F1 for each composition is shown in bold. 94
Table 5.7 : Number of sentences per derivational boundary count 95
Table 5.8 : Labeled argument labeling scores on other languages for different subunits and composition functions, Best F1 for each composition and language is shown in bold. Best F1 that do not require oracle information is shown in italics. 96
Table 5.9 : Apriori integration results of *char3*, *oracle* and *word*. First three rows are provided as a reference. 97
Table 5.10 : Ensemble of base models via averaging. Models in italics are provided as references. 97
Table 5.11 : Ensemble of base models via stack generalization. Models in italics are provided as references. 98
Table 5.12 : Post integration results for other languages 99
Table 5.13 : **Turkish:** Label errors made by each unit (bi-LSTM) composition. Best precision (P), recall (R) and F1 scores are given in bold. 101

Table A.1 : Thematic Roles 121
Table A.2 : Adjunct Semantic Roles in Turkish PropBank. SR: Semantic Role, Exp: Explanation 122
Table A.3 : Types and Definitions of Features 123
Table A.4 : **Finnish:** Label errors made by each unit (bi-LSTM) composition 124
Table A.5 : **German:** Label errors made by each unit (bi-LSTM) composition 125
Table A.6 : **Spanish:** Label errors made by each unit (bi-LSTM) composition 126
Table A.7 : **Catalan:** Label errors made by each unit (bi-LSTM) composition 127
Table A.8 : **Czech:** Label errors made by each unit (bi-LSTM) composition 128

Table A.9 : Hyperparameters for single unit experiments 129

LIST OF FIGURES

Page

Figure 1.1 : Different syntactic realizations of *evacuation of villages* and *returning to the villages* event. 1
Figure 1.2 : Meaning representations of *I have a dog* with FOL (first row), semantic network (second row) and semantic-frame (third row) based approaches 2
Figure 1.3 : AMR annotation for "The dog ate the bone that he found". Concept and argument definitions are taken from PB. *e,d,b*: symbols; **eat-01** and **find-02**: concepts; ARG1-of: inverse relation .. 4
Figure 1.4 : Turkish PropBank Construction Workflow 6
Figure 2.1 : Cornerstone Software Adjusted for Turkish 14
Figure 2.2 : Example TNC Query: "sev-iş-tir* (to make someone to make love with someone)" 15
Figure 2.3 : Causative derivation of transitive verb *giy (to wear)*. (a) Case marking of arguments for root and causative (b) Labeling semantic roles of giy *(to wear)* (c) Labeling semantic roles of giy-dir *(to make sb. wear sth.)* 24
Figure 2.4 : Argument suppression example for reflexive verb *sakla-n (to hide himself/herself)* 27
Figure 3.1 : The argument *the kid* is shared among the predicates *run*, *go* and *fall down* in the sentence *The kid fell down while going home by running*. SUBJECT links shown with dotted lines refers to deep dependency links. 31
Figure 3.2 : Semantic association of *ekonomik (economic)* to verb *büyü (to grow)* instead of noun *büyüme(growth)*. 32
Figure 3.3 : Semantic annotation of adjective *boşaltılan (discharged)* derived from *boşal (to empty)*. 33
Figure 3.4 : Analysis of sentence *Man relaxed after making everyone laugh* with derived adverb *güldürünce (when sb makes sb laugh). Adam (man)* annotated with *ArgA:causer*, *herkes (everyone)* with *Arg0:laugher* defined by *gül.01 (laugh.01). güldürünce* is labeled with *ArgM-TMP: temporal adjunct Adam (man)* with *Arg1: thing relaxing* for the verb *rahatla.01 (relax.01)* 34
Figure 3.5 : Semantic structure of sentence *I didn't understand the things spoken in the class. Konuş (to speak)* first transforms into adjective, then into noun. *Derste* is labeled as locative argument of *konuş.01*; derived noun *konuşulanları* (*things that are spoken*) as Arg1 of *anla.01 (understand.01)* 34
Figure 3.6 : Semantic arguments of *morpheme* frameset "xlAn.01". 35
Figure 3.7 : Semantic roles of the copula "ol" *(to be)*. 35

Figure 3.8 : Interface of verb sense disambiguation task 38
Figure 3.9 : Test Question Preparation View with *Reason* and *Passed Review* fields. Checkbox design allows marking of multiple answers for test question. 42
Figure 3.10: Monitoring View of Test Questions: *Missed* and *Contested* indicates the ratio of missing/contesting the test question. *On/Off* Button can be used for including/excluding the test questions. 43
Figure 3.11: Judgment per Contributor 44
Figure 3.12: Judgment vs Confidence. The vertical red line marks the confidence level 0,70 44
Figure 3.13: Number of questions that fall into the four different agreement intervals 47
Figure 3.14: Illustration of R-XXX in the sentence “Whatever you say, it will happen." 51
Figure 3.15: Histogram of predicate senses in training data 56
Figure 4.1 : PI, PD, AI and AC steps in SRL pipeline 60
Figure 4.2 : Data size versus PD accuracy 69
Figure 4.3 : Data size versus argument labeling scores 69
Figure 4.4 : Effect of data size on overall scores 70
Figure 4.5 : Data size versus fully correct propositions 70
Figure 4.6 : Lemma vs Error 73
Figure 4.7 : PD error versus sentence length 74
Figure 4.8 : Argument labeling error versus sentence length 75
Figure 4.9 : Distance to verb vs argument labeling error 76
Figure 5.1 : Vocabulary growth with respect to corpus size. 82
Figure 5.2 : Out-of-vocabulary rate with respect to vocabulary size. OOV rate goes below 40% only when a vocabulary of size 10^8 words is used. . 83
Figure 5.3 : Bidirectional LSTM model for SRL 85
Figure 5.4 : Internal structure of an LSTM unit 86
Figure 5.5 : Order in inflectional and derivational morphology 89
Figure 5.6 : Apriori integration of multiple subword units 91
Figure 5.7 : Stack Generalization 92
Figure 5.8 : Precision, Recall and F1 differences per role 102
Figure 5.9 : Comparison of statistical and neural models wrt sentence length 103
Figure 5.10: Mistakes by statistical and neural models wrt long range dependencies 103
Figure 5.11: PD accuracy versus datasize 104

BUILDING OF TURKISH PROPBANK
AND
SEMANTIC ROLE LABELING OF TURKISH

SUMMARY

Understanding human language has been a dream of manhood for more than a decade. Although early science fiction movies have predicted that dream would have come true by now, it has not. Ambiguity in meaning, need for common sense knowledge and the variety in sentence structures are only a few of the obstacles among many on our way to understanding language. Despite many attempts to disambiguate word meaning, analyzing language structure and modeling common sense knowledge to reach this goal, natural language understanding remains as an open research area with many subfields.

In this thesis, we are interested in one its subfields: shallow semantic parsing or semantic role labeling (SRL). SRL aims to dissolve the understanding problem into identifying action/event bearing units and their participants. In that way, independent from the structure of the sentence, the same representation can be produced, *(e.g. "Economy grew by 5%" and "The growth of the economy was 5%" or "The window broke" and "Stone broke the window")*. The output representations of this task can benefit other natural language understanding tasks such as information retrieval, sentiment analysis, question answering and textual entailment.

In order to perform this task a resource that contains the meanings of action/event bearing units (in our case verbs) and their frequent participants, named Proposition Bank (PropBank), should be created to guide the machine learning techniques. Unfortunately creating such a resource requires a large amount of time, budget and experts. Therefore has not been possible for many languages including Turkish. In this thesis we aim to address this issue by incorporating crowd intelligence into the construction workflow. We design a novel workflow that requires minimum number of experts with linguistic knowledge. They have been employed for (1) the first crucial step, where semantic frames are manually created, (2) supplying quality control mechanism by labeling a small amount of questions and (3) double checking the answers of crowdtaskers when taskers did not agree on an answer. Other challenges to create such a resource are posed by rich morphology of Turkish. To address this extreme production of new words that cause theoretically infinite number of action bearing units, we propose to exploit the semantic knowledge that are acquired by root verbs composed with regular morphosemantic features such as case markers. We evaluate our overall approach for building of Turkish PropBank by various inter-annotator metrics and show that our resource is of high quality.

Though creating a resource is crucial, it is not enough for automatic labeling of semantic roles. Second part of this thesis focuses on building such automatic methods that are suitable for Turkish language. For that purpose, we adopt a system that uses a deterministic machine learning model based on linguistic features designed mostly

for high-resource, morphologically poor languages. However Turkish language poses the following challenges: (1) significant amount of out of vocabulary words (words that have not been seen in the dictionary) (2) small number of training instances and (3) high syntactic variance among predicates and their arguments. These issues cause very sparse features that complicate the learning process of the statistical system. We address these challenges by (1) designing better features that exploit the regularity of morphosemantics, thus not as sparse as previous ones; and (2) taking advantage of pretraining on unlabeled data, in other words, exploiting prior knowledge on Turkish words that have been learned through word embeddings. We show that our approach yields to the first robust Turkish SRL system with an F1 score of 79.84. Our experiments with training data size and the features show that (1) morphosemantic features are vital for Turkish SRL; (2) a reasonable SRL system can be trained with proposed features on 60% of the available data; (3) performance greatly degrades in the absence of high-level syntactic features and (4) continuous features model complex interactions between information levels and lead to further improvement in the scores.

Although the statistical SRL system has been shown to be successful in the presence of gold tags, it suffers from accumulating errors of external NLP tools that are required for feature extraction. To address this problem, we introduce a neural SRL system that employs bi-directional long-short-term-memory (LSTM) units to operate on subword units which do not require syntactic preprocessing (or only minimal). Unlike previous techniques that use pretrained word embeddings, the proposed model generates a word embedding by composing the subword units. Available subword composition techniques did not make any distinctions between morphology types. In order to distinguish derivational morphology from inflectional morphology, we propose a linguistically motivated composition technique and systematically analyze the effect of subword and composition types. We show that (1) character based models with bi-LSTM composition perform similar to models that use morphological information for languages with poor morphology, whereas at least 3 percentage point drop is observed on F1 scores for morphologically rich languages and (2) linguistically motivated composition method surpasses other techniques for Turkish SRL. We evaluate various techniques to combine multiple subword units in order to test whether subwords learn complementary features for argument labeling. We show that character and char-trigram combination improves the scores in all cases, whereas combining character with morphology do not help to most languages with rich morphology, suggesting that characters do not capture any information that is not already in embedded in morphological models. Finally all resources are made accessible to encourage researchers to work on Turkish language.

TÜRKÇE ÖNERME VERİ TABANININ OLUŞTURULMASI VE TÜRKÇENİN GÖREV ÇÖZÜMLEMESİ

ÖZET

Doğal dili anlamak, uzun süredir insanlığın hayallerini süslemektedir. Eski bilim kurgu filmleri, bu rüyanın şimdiye kadar gerçekleşmiş olacağını öngörse de, henüz gerçeklememiştir. Doğal dili anlamanın halen çözülememiş sorunlar arasında olmasının temel nedenlerini şu şekilde sıralayabiliriz: dildeki belirsizlikler, bağlamdan kaynaklanan sorunlar, gerçek dünya ve sağduyu bilgisinin gerekliliği, sözcük ve tümce yapılarındaki farklılık. Dolayısıyla doğal dili anlama çalışmaları, bu sorunları çözmeyi amaçlayan ayrı araştırmalarla devam etmektedir.

Bu tez çalışmasında yüzeysel anlam ayrıştırıcı ya da diğer adıyla anlamsal görev çözümlemesine (AGÇ) odaklanılmıştır. AGÇ, doğal dili anlama işini, tümcelerden eylem içeren birimlerin ve bunların ögelerinin çıkarılmasına indirgemektedir. Böylece tümcenin yapısından bağımsız olarak, farklı yapılardaki tümceler için aynı anlamsal gösterim biçimi elde edilecektir. Örneğin *Ekonomi %5 oranında büyümüştür* ve *Ekonomideki büyüme %5'tir* veya *Cam taşla kırıldı* ve *Taş camı kırdı* tümcelerinin anlamsal gösterim biçimleri aynı olacaktır. Anlamsal görev çözümlemesinin çıktıları, makine çevirisi, otomatik soru yanıtlama ve duygu analizi gibi değişik doğal dil işleme alanlarına girdi olarak verildiğinde sonuçları iyileştirildiği gözlemlenmiştir.

Anlamsal görev çözümlemesini gerçekleyebilmek için, makine öğrenme yöntemlerini yönlendirmek üzere eylem içeren birimlerin (Türkçe için yüklemlerin) anlamlarını ve ögelerini içeren bir kaynak, diğer bir deyişle veri tabanı, gerekmektedir. Bu veri tabanına yayınlarda Önerme Veri Tabanı ya da PropBank adı verilmektedir. Böyle bir veri tabanı oluşturmak uzun zaman, büyük bütçe ve çok sayıda dil uzmanı gerektirmektedir. Bu nedenle Türkçe için önerme veri tabanları henüz oluşturulmamıştır. Bu tezde, yukarıda bahsedilen sorun, topluluk bilgisini önerme veri tabanının oluşturulması sürecine katılarak çözülmüştür. Uzman sayısını en az olacak şekilde tasarımı yapılan yeni iş modeli, uzmanlardan yalnızca şu durumlarda yararlanmaktadır: (1) Önerme Veri Tabanının ilk ve önemli adımı olan anlamsal görev çerçevelerinin oluşturulması, (2) kalite kontrol sürecinde belli miktarda soru ve yanıtın elle işaretleme ve (3) işaretleyicilerin üzerinde anlaşamadıkları yanıtların doğru olanlarına karar verme aşamasında. Önerme Veri Tabanının oluşturulmasında karşılaşılan diğer bir zorluk ise Türkçenin eklemeli dil olması, Türkçedeki eklerin çok sayıda olması ve Türkçe sözcüklerin peş peşe çok sayıda ek alması dolayısıyla, Türkçenin kuramsal olarak sonsuz sayıda eylem içeren sözcük üretebilmesidir. Bunun için tüm eylem içeren türetilmiş sözcüklerin, kök çerçevesi kullanılarak karşılanmasına karar verilmiştir. Bu yaklaşımla etiketlenen Önerme Veri Tabanının yüksek nitelikli olduğu çeşitli işaretleyici uzlaşması ölçme yöntemleri kullanılarak kanıtlanmıştır.

Bu tezin ikinci bölümünde, Türkçe AGÇ'ye uygun makine öğrenme yöntemlerinin geliştirilmesi üzerinde durulmuştur. Bu amaçla sonucu kesin (deterministik) bir

makine öğrenme modeli olan lojistik regresyon sınıflandırıcısı kullanılmıştır. İlk olarak, diğer dillerin anlamsal görev çözümlenmesi için tasarlanmış öznitelikler kullanılmış, fakat başarımlarının yetersiz olduğu gözlemlenmiştir. Bunun nedenleri şöyle açıklanabilir: (1) derlem dışı sözcüklerin çokluğu (2) eğitim kümesinin küçük olması, (3) eylem ve ögelerinin sözdizimsel farklılıklarının yüksek olması. Bu özellikler, çıkarılan özniteliklerin seyrek olması nedeniyle istatistiksel sistemin anlamsal görevler hakkındaki kalıpları öğrenememesine neden olmaktadır. Bu sorunları azaltmak amacıyla, (1) Türkçe diline daha uygun olan biçim bilimine dayalı öznitelikler (özellikle adın durumları), (2) büyük etiketsiz veri kümesinde eğitilmiş sözcük vektörlerine dayalı öznitelikler kullanılmış ve bu özniteliklerin AGÇ'nin başarımını artırdığı gözlemlenmiştir. Böylece ilk yüksek başarımlı (79.84 F1 puanlı) Türkçe AGÇ sistemi geliştirilmiştir. Deneylerimiz (1) biçim anlamsal özniteliklerin Türkçe AGÇ için önemini; (2) tasarlanan sistemin eğitim verisinin yalnızca %60'ını kullanarak, anlamlı sonuçlar üretilebileceğini; (3) bağlılık ağacı ve söz dizimsel sınıf bilgisine dayalı özniteliklerin yokluğunda performansın azımsanmayacak şekilde düştüğünü ve (4) sürekli özniteliklerin bilgi seviyeleri arasındaki etkileşimi modelleyerek başarıyı artırdığını göstermiştir.

İstatistiksel sistemin, sözcüklerin gerçek etiketlerinin bilindiği durumda başarılı olduğu gösterilmişse de, bu etiketlerin bilinmediği durumda peş peşe kullanılan doğal dil araçlarının her birinden kaynaklanan hataların birikmesi dolayısıyla performansı düşmektedir. Bu nedenle, araçlara en az düzeyde ihtiyaç duyan, çift yönlü LSTM birimlerinin alt sözcükleri işlemesine dayanan bir yapay sinir ağı yöntemi önerilmiştir. Eğitilmiş sözcük vektörleri kullanan önceki yöntemlerin tersine, önerilen yöntem alt sözcükleri çeşitli fonksiyonlarla birleştirerek sözcük vektörü yaratmaktadır. Varolan birleştirme yöntemleri biçimbirimsel farklılıkları göz önüne almamaktadır. Bu nedenle yapım ve çekim eklerinin ayrı ayrı birleştirildiği farklı bir yöntem sunulmuştur. Alt sözcük birimleri ve birleştirme fonksiyonları sistematik olarak analiz edilerek, etkileri ölçülmüştür. (1) Yalnızca karakter bilgisi kullanan modellerin, zayıf üretme yetenekli diller için biçimbirimsel bilgi kullanan modellerle benzer sonuçlar verdiği fakat üretim bakımından zengin dillerde biçimbirimsel bilginin başarımı en az yüzde 3 puan artırdığı (2) önerilen birleştirme yönteminin öncekilerden daha başarılı olduğu gösterilmiştir. Alt sözcüklerin AGÇ için tamamlayıcı özellikler öğrenip öğrenmediğinin sınanması için birden çok alt sözcük tipi çeşitli tekniklerle birleştirilmiştir. Karakter ve karakter üçlülerinin birleştirilmesinin her durumda başarımı artırdığı gözlemlenmiş, fakat biçimbirimsel bilginin karakterle birleştirilmesinin, üretken diller birçok dile yardımcı olmadığı görülmüştür. Bu bulgu, karakter modellerinin, söz konusu diller için, zaten biçimbirimsel modellerde olmayan herhangi bir bilgiyi yakalayamadığını düşündürmektedir. Son olarak, araştırmacıların Türkçe dili üzerinde çalışmasını özendirmek amacıyla tüm kaynaklar erişilir biçimde tüm araştırmacılara sunulmuştur.

1. Introduction

There are many ways to express the same thing. Sentences given in Fig. 1.1 [1] are all different but they all express that two main events *evacuation of villages* and *returning to the villages* occur. Ideally, we would want these sentences to have approximately

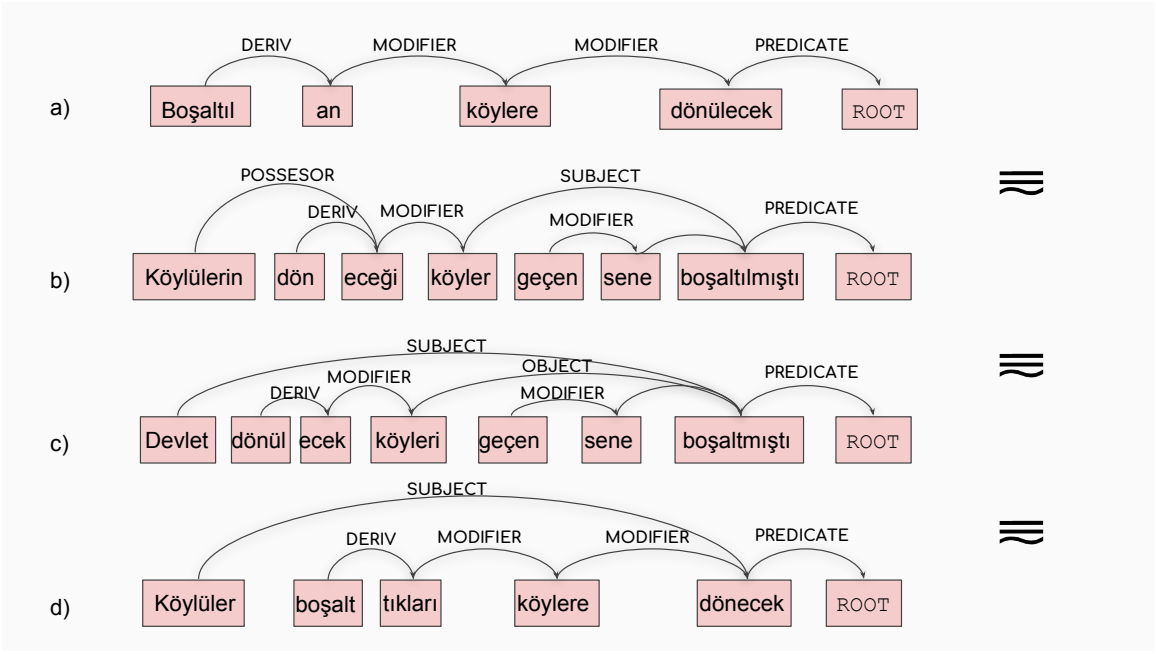


Figure 1.1 : Different syntactic realizations of *evacuation of villages* and *returning to the villages* event.

the same representations. However, despite the ability of syntactic parsers to extract a lot of useful information from sentences, they are not able to produce the desired **Meaning Representations (MRs)**.

Researchers in natural language processing field have introduced different frameworks, so called **Meaning Representation Languages (MRL)**, that specify the syntax and semantics of MRs. Fig. 1.2 shows sample MRs using three commonly used languages for the sentence *I have a dog*. First-Order Logic (FOL) [1] in the first, semantic

[1]Glossary for Fig. 1.1: a) Boşaltılan köylere *(to the evacuated villages)* dönülecek *(will be returned)*. b) Köylülerin döneceği köyler *(the village where the villagers will return to)* geçen sene boşaltılmıştı. *(were evacuated last year)* c) Devlet *(the government)* dönülecek köyleri *(the villages to be returned)* geçen sene *(last year)* boşaltmıştı *(had evacuated)*. d) Köylüler *(the villagers)* boşalttıkları köylere *(to the villages they evacuated)* dönecek. *(will return)*

network in the second [2], and frame-semantics in the last [3] row. In this thesis, we

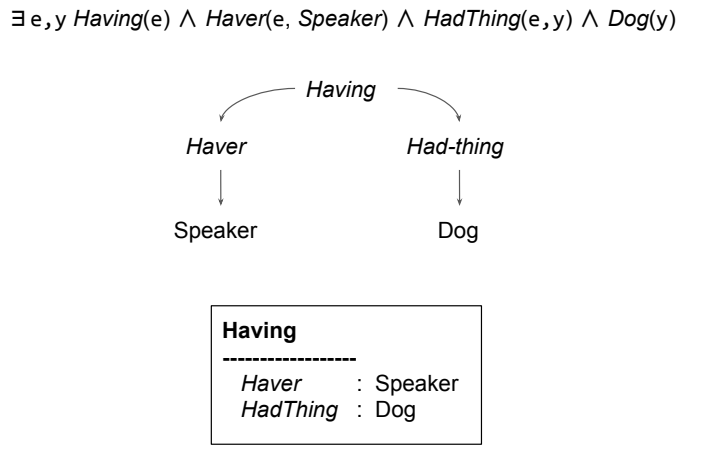


Figure 1.2 : Meaning representations of *I have a dog* with FOL (first row), semantic network (second row) and semantic-frame (third row) based approaches

focus on frame-semantic languages which are easily linked to syntactic input, easily set up for new properties and relations and good at handling missing values. From this point on, we use the word **lemma** a.k.a citation form, as a pairing of ortographic form with its meaning; and **lexicon** as a finite list of lemmas. We frequently use root verbs as lemmas *(e.g., bul for buldum, bulundu, bulunan)*. Our focus is analyzing the semantics of events. Therefore, whenever we use frame, we refer to a verb frame. Events are analyzed via predicate-argument structures. Predicate means the surface form *(e.g., buldum, bulundu, bulunan)* of the event-bearing word, which will be linked to its lemma *(e.g., bul)*. Argument refers to predicate's complementary and its relation to its predicate is named as **semantic role**.

Common frame-semantics approaches are **FrameNet (FN)** [4], **VerbNet (VN)** [5], **PropBank (PB)** [6] and recently introduced **Abstract Meaning Representation (AMR)** [7]. These resources differ in type of semantic roles they use and type of additional information they provide. FN is a network, built around the theory of **semantic frames**. This theory describes a type of event, relation, or entity with their participants which are called **frame elements (FEs)**. All predicate lemmas in the same semantic frame share one set of FEs. A sample sentence annotated with FN, VN and PB conventions respectively, is given in Example 1. The lemma *buy* belongs to *Commerce buy*, more generally *Commercial transaction* frame of FN which contains *Buyer*, *Goods* as core frame elements and *Seller* as a non-core frame element as in Example 1. FN also provides connections between semantic frames like inheritance,

hierarchy and causativity. For example the frame *Commerce buy* is connected to *Importing* and *Shopping* frames with *used by* relation. FN is originally built as a lexicon of verbs, however with emergence of SRL task a corpus has been annotated with its semantic frames.

Example 1 [Jess]$_{\text{Buyer-Agent-Arg0}}$ bought [a coat]$_{\text{Goods-Theme-Arg1}}$ from [Abby]$_{\text{Seller-Source-Arg2}}$

Syntax: Agent V Theme {From} Source

Contrary to FN, VN is only a hierarchical verb lexicon, that contains categories of verbs based on Levin Verb classification [5]. For instance *buy* is contained in *get-13.5.1* class of VN, among with the verbs *pick*, *reserve* and *book*. Members of the same verb class share same set of semantic roles, referred to as **thematic roles**. In addition to thematic roles, verb classes are defined with different possible syntaxes for each class. One possible syntax for the class *get-13.5.1* is given in the second line of Example 1. PB makes use of predicate specific coarse-grained semantic roles *(e.g., Arg0, Arg1 and Arg2 for Buyer, Goods and Seller)* while FN specifies very fine-grained semantic roles *(e.g., Buyer, Goods and Seller for predicate "buy").*

AMR makes heavy use of PB frames and represents sentences as a single rooted, labeled, directed semantic graphs. In addition to semantic roles defined by PB, it incorporates unspecific semantic relations (*e.g. age, name, unit, scale*) and important linguistic phenomena like coreference, modality, copula and negation. It aims to assign the same AMR to the sentences with the same basic meaning even though the predicates are different. For example *he delivered the package*, *the deliverance of the package* or *the package was delivered* are aimed to have the same AMR. Since this formality abstracts away from syntax, it has recently inspired many studies [8] and *Meaning Representation Parsing* shared task in SemEval-2016 [9]. An example annotation is provided in Fig. 1.3.

VN defines possible syntaxes for each class of verbs. However, due to free word order and excessive case marking system, syntactic information is already encoded with case markers in Turkish. Thus the structure of VN does not fit well to the Turkish language. FN provides a high level of abstraction by allowing a class of related verbs to share role labels while PB uses argument labels (Arg0, Arg1, etc.) that are meaningful

```
(e / eat-01
  :ARG0 (d / dog)
  :ARG1 (b / bone :quant 1)
        : ARG1-of (f / find-01)
              :ARG0 d)))
```

Figure 1.3 : AMR annotation for "The dog ate the bone that he found". Concept and argument definitions are taken from PB. *e,d,b*: symbols; **eat-01** and **find-02**: concepts; ARG1-of: inverse relation

only with regard to a specific predicate [2]. FN defines richer relations between verbs, but the frame elements are extremely fine-grained and building such a comprehensive resource requires a great amount of manual work for which human resources are not currently available for Turkish. Contrary to FN, PB greatly simplifies the semantic roles but neither defines relations between verbs nor guarantee consistency among labels. Even though the formalisms among mentioned resources differ, there is a clear relation between FN and PB as shown by [10]. AMR is the closest formalism to the ideal *interlingua* representation however it requires a PB and annotation of variety of linguistic information like co-reference and named entity.

More general semantic roles and high annotation consistency reported by English PB have inspired many other PBs such as Hindi [11], Chinese [12], Arabic [13], Finnish [14] and Portuguese [15] and have been used as the standard annotation scheme for SRL related shared tasks for the last decade [16–19]. Furthermore, it has been the skeleton of the promising semantic resource AMR. Taking all these facts into consideration, PB formalism have been chosen as our standard annotation scheme.

Below, we provide an example sentence containing a *selling* event, annotated with PB scheme. Here, *lemma.i* refers to the i_{th} meaning of the predicate *lemma*. Its semantic roles/arguments are shown with symbol Aj, where j refers to the argument class (a number or a modifier).

sat.01	**sell.01**	
A0: Satıcı	A0: Seller	[Ayşe]$_{A0}$ [elbisesini]$_{A1}$ [Fatma'ya]$_{A2}$ [sattı]$_{sat.01}$
A1: Satılan	A1: Thing Sold	[Ayşe]$_{A0}$ [her dress]$_{A1}$ [to Fatma]$_{A2}$ [sold]$_{sell.01}$
A2: Alıcı	A2: Buyer	

[2] Although there are no consistent generalizations across verbs in PB, Arg0 is used for actor, agent, experiencer or cause of the event; Arg1 represents the patient, if the argument is affected by the action, and theme, if the argument is not structurally changed.

The automatic process of identifying predicate-argument structures and assigning meaningful labels to them (similar to the example above) is named **Semantic Role Labeling (SRL)**. It is considered as an important task, and has gathered attention from the NLP community for some time due to its potential for utilizing high level natural language understanding problems. There have been two shared tasks at CoNLL 2004 and 2005 [16, 17], where participants were asked to assign semantic roles to syntactic constituents of predicates. In CoNLL 2008 and 2009 [18, 19], it was slightly modified to include nominal predicates [20] and to label nodes of dependency trees rather than constituents. More recently broad-coverage semantic dependency parsing, a task very close to SRL in spirit, has been tackled as part of SemEval 2014-Task 8 [21] and SemEval 2015-Task 18 [22]. In conclusion, it is seen as a gateway to real understanding of natural language and therefore is crucial to natural language understanding field. Moreover, it has been shown to provide benefits to complex natural language processing tasks such as information retrieval [23–25], question answering [26, 27], textual entailment [28] and machine translation [29–32].

1.1 Statement of the Problem

Turkish does not have a FN, PB or a similar semantically interpretable resource that defines SRL as task. In literature the common practice for building a PB is first to create semantic frames for predicate lemmas then to annotate predicates and their arguments in the corpus with their corresponding senses and roles. All of these processes are manually performed and require large numbers of annotators, a long time and a big budget. To address this issue, transferring knowledge from resource-rich languages to resource-poor languages by means of parallel corpora is proposed by [33]. However automatic methods suffer from translation shifts, paucity of parallel corpora, predicate mismatches and word alignment problems (greater degree of alignment errors are expected for English-Turkish language pair due to rich derivational morphology of Turkish [34]).

Our thought was while some of the steps in building a PB need linguistic expertise, some can be performed without fully incorporating experts. Thus we have designed a workflow as shown in Fig. 1.4 that incorporates experts only when necessary. The first step, **framing**, includes making important decisions on linguistic phenomena

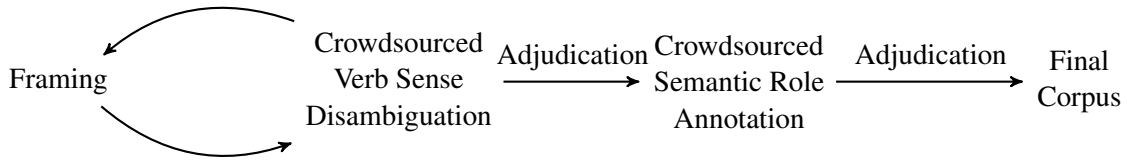


Figure 1.4 : Turkish PropBank Construction Workflow

such as number of verb senses and type of arguments a verb governs. Therefore frame files need to be created by experienced annotators by examining large amounts of sentences for each predicate ensuring a solid empirical grounding on expected semantic roles. After the framing step, predicates in the corpus should be annotated with their corresponding rolesets which is referred to as **verb sense disambiguation (VSD)**. After VSD step, the verbs with low agreement scores should be presented to the adjudicator (a more experienced annotator) to decide on the correct sense in order to minimize the errors in the generated resource. Finally, semantic annotations should be added on top of the syntactic annotations as a separate layer where semantic role labels are assigned to nodes in the dependency tree, referred to as **semantic role annotation (SRA)**.

After building of the necessary resource, an automatic method that can learn from the annotated corpus is necessary. Common approach in the field is to divide this task into subtasks, namely as predicate identification (PI), predicate sense disambiguation (PSD), argument identification (AI) and argument classification (AC), then train local classifiers for each task. However, current statistical methods in the literature use features that are tailored for morphologically-poor languages, therefore not suitable for Turkish language; and apply their methods on large number of training instances and suffer less from out of vocabulary (OOV) words. Furthermore, they rely on engineered linguistic features that are supplied by external NLP tools. This pipeline approach causes error accumulation that has severe effects on the end task.

1.2 Main Contributions of the Thesis

Our contributions in this thesis are three-folds:

1. We present the first semantically annotated corpora **Turkish PropBank**. We believe it will not only provide benefit to local NLP community for novel research in

the Turkish language but also the global NLP community for building better SRL systems that can handle languages with complex morphology. To encourage all researchers to work on the Turkish SRL problem, we release frameset lexicon, annotated semantic layer on IMST and IMST-UD and crowdsourcing designs from http://turkishpropbank.github.io/.

2. We train the first SRL system on Turkish PropBank that employ language specific features. We further improve the performance by incorporating continuous features that are composed of pretrained word embeddings. Source code of all systems and trained embeddings are also distributed from the project page.

3. We build a neural SRL model performing on subword units in order to avoid feature engineering and using of external NLP tools and show that:

   (a) character based models with bi-LSTM composition perform similar to models that use morphology for morphologically-poor languages, whereas a large drop occurs on F1 scores for morphologically-rich languages,

   (b) linguistically motivated composition method surpasses other combining techniques,

   (c) statistical system with engineered features is hard to beat by a neural model.

1.3 Organization of the Thesis

This thesis is organized to guide readers through the journey of a Turkish sentence, (*e.g., "Yere düşen elmaları topladım."*) converted into its meaning representation, in our case, a collection of predicate argument pairs, e.g., `topla.02` (`toplayan`: Ben, `toplanan`: elma); `düş.01`(`düşen`: elma, `varış yeri`: yer). This journey is presented in two parts. The first part focuses on building of the **Turkish PropBank** that consists of building of the semantic lexicon that contains predicate senses and arguments they govern (see Chapter 2) and creating a semantically annotated corpora (see Chapter 3). Second part, comprising Chapter 4 and Chapter 5 discusses automatically extracting meaning representations, in other words semantic role labeling. In Chapter 4, we employ statistical machine learning techniques that use discrete and continuous features. Chapter 5 presents our approach to neural end-to-end SRL for morphologically-rich languages. Each chapter is designed to be self-contained

with a dedicated background and related work section, due to the methods being diverse. Finally Chapter 6 concludes with a summary of contributions, key findings and detailed future work. Contents of chapters are as follows in detail:

Chapter 2 presents our method for framing the verbs of Turkish PropBank. We discuss the manual framing process by experts with the help of publicly available dictionaries, corpora and guiding morphosemantic features such as case markers. Then, we present a systematic way of framing for challenging cases such as light verbs, multiword expressions and derived verbs. We define the semantic roles and motivate for root verbs only policy. In conclusion, a new lexicon of Turkish verbs with 773 verb frames and 1285 senses is constructed.

Chapter 3 discusses the complete annotation framework and describes how crowdsourcing has been used to create a semantically annotated corpora. It first presents annotation challenges introduced by Turkish and the properties of IMST we have built the Turkish PropBank upon. It continues with crowdsourcing of verb sense disambiguation and semantic role annotation. Finally, we demonstrate the quality of the created resource by means of various annotation agreement measures.

Chapter 4 demonstrate our statistical approach for automatic Turkish semantic role labeling. We train separate logistic classifiers for PD, AI and AC steps that uses separate set of language specific features and distributional semantics. We evaluate our methods on the resource we have created in Chapter 2 through 3. We carry out experiments to investigate the effect of data size, morphosemantic features; necessity of information level of features (lexical, positional, morphological, syntactic and semantic) for a robust SRL and contribution of continuous features. We perform an error analysis on label predictions and try to determine the source of errors for different steps.

Chapter 5 aims to (1) address the error accumulation problem caused by pipeline approaches, (2) eliminate language-specific feature engineering and (3) reduce dependencies to lower level NLP tools such as morphological analyzers/disambiguators and dependency parsers. For that purpose, we present a neural sequence tagger based on LSTMs. We compare performances of sentence encoders that use words, subwords and various combination of different units and different composition techniques.

Furthermore we experiment with integrating knowledge from multiple subword units and test whether there exists any complementary units. We carry out analysis on label predictions of different units and compare them with results of statistical system described in previous chapter.

2. Framing of Turkish

Framing can be defined as creating **semantic frames** for argument governing units, *generally verbs*, by following the guidelines determined by the semantic formalism. It is the first and the most important step for creation of a semantically interpreted resource. The errors introduced in framing process may accumulate and may significantly reduce the accuracy and reliability of the semantic corpora and semantic role labeling task. It requires making important decisions on linguistic phenomena such as number of verb senses, multiword expressions, light verb constructions and type of arguments a verb governs. In this chapter, we first give background and related work on incorporating morphosemantics into building of a semantic lexicon (Section 2.1). Section 2.2 presents the modified framing tool and the framing guidelines for Turkish Proposition Bank. Then in Section 2.3, we investigate how the semantical information supplied by morphemes can be used during framing process.

2.1 Background and Related Work

In study by Agirre [35], the authors discuss the suitability of PropBank model for Basque verbs. In addition to semantic role information, case markers that realize these roles are included in the verb frames. A database that contains syntactic/semantic subcategorization frames were readily available for 100 verbs. They have used this information for tagging the arguments automatically in Basque Dependency Treebank (BDT), for only 10 most frequently used verbs. Later Aldezabal [36] shows the progress of the Basque PropBank project and defines a more detailed methodology and an annotation tool. These studies show that including case markers in Basque PropBank as a morphosemantic information provider can be useful for automatic tagging of semantic roles for Basque language which has 11 case markers.

Hawwari and colleagues [37] present a pilot study for building Arabic Morphological Pattern Net, that aims to represent a direct relationship between morphological patterns and semantic roles for Arabic language. Authors experiment 10 different patterns and

2100 verb frames and analyze the structure and behavior of these Arabic verbs. The authors state that the results encourage them for a more comprehensive study.

Furthermore, there are studies on exploiting morphosemantics in WordNets for different languages. Fellbaum [38], manually inspects WordNet's verb-noun pairs to find one-to-one mapping between an affix and a semantic role for English language. For example the nouns derived from the verbs with the suffixes $-er$ and $-or$, like *invent-inventor* usually results as the agents of the event. However, it is stated that only two thirds of the pairs with this pattern could be classified as agents of the events. More patterns are examined and the regularity of these patterns are shown to be low for English language. In another work [39], authors propose a methodology, on exploiting morphosemantic information in languages where the morphemes are more regular. They perform a case study on Turkish, and propose application areas both mono-lingually and multi-lingually, such as globally enriching WordNets and auto detecting errors in WordNets. In a similar work [40], morphosemantic information is added to Romanian WordNet and the proposed application areas in [39] are examined and shown to be feasible.

Previous studies based on building Basque PropBank focus on the building process of Basque PropBank, rather than analysis of the regularity of case markers and the relation between semantic roles and case markers. Furthermore, the study related to building Arabic Morphological Pattern Net, aims to build a separate dataset and map it to other resources such as Arabic VerbNet, WordNet and PropBank. WordNet has rich cross-language morphosemantic links however it does not list all arguments of predicates, thus its structure is not convenient for NLP tasks like semantic role labeling.

2.2 Method

Semantic frames are created in order to guide the annotation process discussed in Chapter 3. They contain a list of framesets, i.e., **coarse-grained** verb/predicate senses, and each frameset contains a list of verb/predicate specific roles, known as arguments or semantic roles, and different syntactic realizations of the verb.The list of "expected arguments" of each roleset is referred to as **core** or **numbered** arguments, and are labeled as *ArgN*, where *N* takes an integer value between 0 and 5.

The common procedure is:

1. investigate a number of sentences that contain the verb to be framed;
2. create roles that are encountered often and/or semantically necessary - repeat this for each sense of the verb;
3. number these roles sequentially from A0 (Arg0) up to A5 (Arg5) as suggested by the PropBank framing guidelines [41].

An example frameset for "çalış" is given in Table 2.1:

Roleset id:	çalış.01	emek harcamak
Roles:		
Arg0:	*emek harcayan kişi*	NOM
Example:		
	Çalışan ilerler, yerinde kalmaz	
Roleset id:	çalış.02	İşi veya görevi olmak, bulunmak
Roles:		
Arg0:	*görevli olan kişi*	NOM
Arg1:	*hangi görevde çalıştığı*	NOM
Arg2:	*görevli olduğu yer*	LOC
Example:		
	Artık diğer otellerde kaç kişi çalışıyor hesaplayın.	
	Arg0: kaç kişi	
	Arg2: diğer otellerde	
	ArgM-DIS: Artık	
Roleset id:	çalış.03	Bir şeyi öğrenmek ya da yapmak için uğraşmak
Roles:		
Arg0:	*emek harcayan kişi*	NOM
Arg2:	*emek harcadığı şey*	DAT
Example:		
	Üç senedir piyano çalmaya çalışıyor.	
	Arg2: piyano çalmaya	
	ArgM-TMP: Üç senedir	

Table 2.1 : Example framesets of "çalış"

This section will discuss the procedures that we have used for deciding on the verbs to be framed and number of senses (rolesets) the verb has. We discuss the most common scenario (root verbs) and then investigate LV and MWE cases. Then we describe the process of deciding on the numbers and types of semantic roles a verb governs.

2.2.1 Framing Tool

For framing purposes, we have adjusted an already available open source software, cornerstone [42]. Cornerstone had been used for building English, Chinese and Hindi/Urdu PropBanks. Due to the close connection between case markers and semantic roles (discussed in Section 2.3.1), we incorporate case marking information of numbered arguments into the framing process. To supply case marking information of the argument, a drop down menu containing six possible case markers in Turkish is added as shown in Fig 2.1.

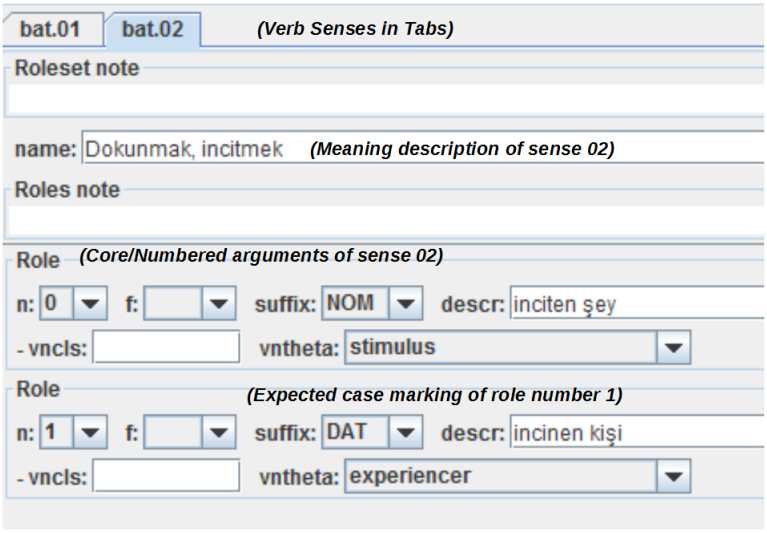


Figure 2.1 : Cornerstone Software Adjusted for Turkish

2.2.2 Distinguishing Senses

English PropBank is constrained to the verbs that are only encountered in the corpus to be annotated. Unlike English PropBank, we have initiated our framing efforts with the list of Turkish root verbs provided by TDK. The reason for our decision the size of our corpus: IMST and its low coverage rate for Turkish verbs. This list consists of 759 root verbs however it contains verbs that are rarely used or have fallen into disuse as the ones shown in Table 2.2. In order to detect those root verbs we have used TNC (Turkish National Corpus), which is a balanced and a representative corpus of modern Turkish with about 50 million words. Its query interface shown in Fig. 2.2 allows regular expressions which is essential for querying verbs that appear in different conjugated forms in unstructured text. We have performed queries on all root verbs and framed them if their frequency count is above 5 in a million words. Overall only 385

1	OG24D1B-2300	bunun yanı sıra geometriyi demokrasiyle	**sevistiren**	"Bellavista-Bellevue" (Güzelyaşam/İyihayat) ya da
2	SE36E1B-3352	tanıtım yazısındaki ihtiras ile melodramı	**sevistiren**	bir öykü gibimanasız manalandırmalar ve
3	KA16B4A-0121	bir araya getirdi. Hepsini de	**sevistirdi.**	sonunda da bu başörtünün hangi
4	QI22C1A-0532	Sıkı sıkı. Yüreklerimizi buluşturacağız, ruhlarımızı	**sevistireceğiz**	önce. Kıpırdamadan duracağız. Ne zama
5	KA16B4A-0121	bir çok delikanlısıyla birkaç kızını	**sevistirdi.**	Yalnız, esmer güzeli, yirmi yaşındaki
6	VI22F1D-4708	bayağı güzel geyiğini çevirmiştik. Protonları	**sevistirecekler**	diye duydum en son. *
7	SE36E1B-3352	ile hüznü, ihtiras ile melodramı	**sevistiren**	bir öykü. Kahramanlarının birbirine estetik
8	SE36E1B-3352	düzey bir tolerans kültürü de	**"sevistiren"**	bir çabayla çatışmasız, çelişkisiz bir
9	UD22C1A-0510	Çocukluğunda Barbie ile öpüşen, Barbieler'ini	**sevistiren,**	bu esnada ailesine yakalanan yetişkinlerin
10	LE41C2A-1360	ve aynı durumdaki insanlarla sahnede	**sevistiriliyormuş."**	Çıkartmalardaki LSD Evet, içimiz yeteri
11	UA16B3A-1065	büyük arzusu Meltem ile Handan'ı	**sevistirmekti.**	İstediği de gerçekleşti. Kadınlar çılgınca

Figure 2.2 : Example TNC Query: "sev-iş-tir* (to make someone to make love with someone)"

of the verbs were found to be above this threshold. Some exemplary root verbs that were excluded from the framing process are given with their frequencies in Table 2.2.

Root Verb	Count	Frequency
eğir (to spin cotton for making thread)	105	2,24
semir (to batten, get fat)	80	1,68
yüksün (to regard someone, something as a burden)	52	1,09
çıv (to be deflected)	24	0,5
evele (to hum and haw)	16	0,34
göynü (to be grieved)	5	0,1
ılga (to run at a gallop - used only for horses without a rider)	5	0,1
çemre (to roll up one's sleeves, trouser legs, or skirts)	4	0,08
ipile (to give a very dim light)	1	0,02
fışılda (to make a swishing or rustling sound)	0	0

Table 2.2 : Excluded root verbs and their frequencies in a million

To decide on rolesets, framers examine a large number of query results (queries with different syntactic realizations of the verb) similar to Fig. 2.2 together with dictionary entries provided by TDK.

The main principle for distinguishing framesets is to check if two sets have different number of arguments, or the number of arguments are the same, but the thematic roles are different. List of thematic roles and their explanations are given in Appendix A.1. In other words, different senses should have different types/numbers of allowable arguments. Therefore, unlike a TDK dictionary definition, distinctions are very coarse, e.g., metaphors and literal meanings are not distinguished. One frameset usually corresponds to several standard dictionary entries.

Two tests suggested by framing guidelines [41] are follows:

- Check if one sense entails the other.
- Check if the set of roles for one sense is a subset of another sense's roles.

2.2.2.1 Light Verbs and Multiword Expressions (MWE)

Turkish is known to be a language that has borrowed many foreign words from other languages such as Arabic, Persian, French and lately English due its multinational historical background. That led to large number of light verbs (LV) and multi word expressions (MWE). LV and MWE are still an active research area for linguists [43], and due to the complexity of this issue annotation of LV and MWE constructions in PropBank has been investigated separately in study [44].

Light verbs are the verbs that cannot stand in the sentence on their own but can occur with another verb or a nominal [45]. Light verb constructions in Turkish are the complex predicates formed by a nominal and one of the light verbs such as ol-, et-, gel-, ver-, dur-, kal-, düş-, bulun-, eyle- and buyur- [43]. Other than Turkish, light verb constructions can also be encountered in many languages such as Japanese, Korean, Persian, English, French and German.

Light verb itself may contribute comparatively light to the meaning or it has no contribution as in 'teşekkür et- (to thank) '. In such cases, where the meaning is mostly conveyed by the nominal, the phrase is treated as a new predicate as (teşekkür_et). In addition, Turkish light verbs are not necessarily light in all uses. Consider the function of the verb et- in the sentence "Üç artı iki beş eder (Three plus two makes five)". Framing process is handled similarly for such verbs as in other root verbs.

Most of the time, MWEs are confused with light verb constructions. In order to avoid discussions, we approach the problem practically, rather than categorizing verbs as LVC or MWE. We either treat such verbs as another sense of the root verb or as a complex predicate. The criterias followed during the decision process are:

- Deviation from the original meaning of the verb root,
- Contribution of nominal to the meaning of the complex predicate,
- The frequency of the complex predicate,

- Being a **fixed** phrase,

In Table 2.3. our framing approach for the verb *ver (to give)* is shown as an example. Second sense has the meaning of *to fix, to establish* as in to give/fix appointment, name or price. Similarly ver.03 is defined as to devote, allocate as in *öncelik vermek (to give priority)*, *emek vermek (to give/devote effort)* and *zaman vermek (to give/allocate time)*. These phrases are not fixed and the contribution of the nominal is not dominant. Hence they are framed with new senses for the root verb. On the contrary, the complex predicates, *söz ver (to promise)*, *izin ver (to allow)*, *kulak ver (to listen carefully)* and *hesap ver (to explain)* are fixed phrases and they have high frequency in TNC corpus. Hence they are determined as separate predicates.

Predicate	Sense	Meaning	Example
ver	ver.01	To transfer	Hediye vermek (Give presents)
ver	ver.02	To fix	Randevu vermek (Give an appointment)
ver	ver.03	To devote, allocate	Öncelik vermek (Give priority)
söz ver	ver.09	To promise	Bana söz ver (Promise me)
kulak ver	ver.12	To listen carefully	Bana kulak ver (Listen to me)

Table 2.3 : Framing of the verb "ver- (to give)"

2.2.3 Semantic Role Numbering

As discussed at the beginning of this chapter, PropBank simplifies semantic roles but does not guarantee consistency among labels, apart from the roles Arg0 and Arg1. It consistently assigns Arg0 to agents or experiencers, and Arg1 to the patient argument, i.e. the argument which undergoes the change of state or is being affected by the action. Derivational morphemes give rise to some exceptional cases that are discussed in Section 2.3.

Although Arg0 and Arg1 are the only labels which are associated with a certain specific semantic content, we have tried to follow the following trend for the other numbered arguments as given in Table 2.4.

In addition to core labels, all predicates can govern a set of general, adjunct-like arguments, identified by the function tags shown in Table 2.5. These arguments are generally named as *ArgM* or *AM*, where *M* is one of these function tags. In order to allow annotation of every constituent surrounding the verb, English PropBank included tags such as *MOD* for modal verbs and *NEG* for verb-level negation though they are

Argument	Thematic Role
Arg0	agent, experiencer
Arg1	patient, theme
Arg2	beneficiary, instrument, recipient
Arg3	source, beneficiary, instrument
Arg4	destination

Table 2.4 : Thematic roles commonly associated with numbered arguments

ADV	Adverbial	LOC	Location
CAU	Cause	LVB	Light Verb
COM	Commitative	MNR	Manner
DIR	Direction	NEG	Negation
DIS	Discourse Connectives	TMP	Time
EXT	Extent	TWO	Verb Reduplication
GOL	Goal	INS	Instrument

Table 2.5 : The complete list of semantic labels for temporary roles

not considered adjuncts [6]. For similar reasons, we include the modifier AM-TWO for annotation of Turkish verb reduplications such as “koşa koşa" *(running running)* (a phrase used for expressing *great eagerness*). Some of the core arguments may correspond to one of adjunct-like arguments in examined sentences. In those cases, that argument is marked as a numbered argument, if it frequently occurs in a corpus and is specific to a particular class of verbs.

2.3 Morphosemantics

In morphologically rich languages, the meaning of a word is strongly determined by the morphemes that are attached to it. The semantical information supplied by morphemes is named **morphosemantics**. Some of these morphemes always add a predefined meaning while some differ, depending on the language. However, only regular features can be used for NLP tasks that require automatic semantic interpretation.

In this section, we focus on two regular Turkish morphosemantic features critical for SRL: case markers and verb derivational morphemes. We hypothesize that these features can be incorporated into construction of semantic resources to help us decrease the manual effort, increase consistency and connectivity of the resource and increase the performance of SRL.

Number of Cases vs Number of Languages

2 cases	3 cases	4 cases	5-7 cases	8-9 cases	10 or more cases
23 languages	9 languages	9 languages	39 languages	23 languages	24 languages

Table 2.6 : Case marking across languages (taken from World Atlas of Language Structures)

	TR	HR
NOM	**Ben** geldim. **I-NOM** come-PAST. **I** came.	**Ági** jött. **Ági** come-PAST **Ági** came.
ACC	Avcı **tavşan-ı** gördü. Hunter **the rabbit-ACC** see-PAST. The hunter saw **the rabbit**.	Látom a **hegy-et**. see-P1s **mountain-ACC**. I see **the mountain**.
DAT	Jack **okul-a** gitti. Jack **school-DAT** go-PAST. Jack went **to school**.	**Ági-nak** adtam ezt a könyv-et. **Ági-DAT** give-P1s-PAST book-ACC. I gave this book **to Ági**.
LOC	**Ankara'da** oturuyorum. **Ankara-LOC** live-P1s-PRES. I live **in Ankara**.	**Budapest-ban** lakom. **Budapest-LOC** live-P1s-PRES. I live **in Budapest**.
ABL	**Annem-den** geldim. **Mother-ABL** come-P1s-PAST. I came **from my mother**.	**Ági-tól** jöttem. **Ági-ABL** come-P1s-PAST. I came **from Ági**.

Table 2.7 : Case marking in Turkish and Hungarian

2.3.1 Case Marking

Declension is a term used to express the inflection of nouns, pronouns, adjectives and articles for gender, number and case. It occurs in many languages such as Arabic, Basque, Sanskrit, Finnish, Hungarian, Latin, Russian and Turkish. Table. 2.6 shows, there are 86 languages with at least 5 case markings. An examplary morphological analysis for the Turkish word "evlerinde" (in his houses) is shown below. In this analysis, ev is inflected with $-ler$ morpheme for plurality, i for third person singular and de for locative (LOC) case.

Example 2 ev (- ler) (-i) (-nde)

ev +Noun+ Pl + P3s + LOC

Even though the languages differ, the same case markers are used to express similar meanings with some variation. In order to exemplify this statement, sentences with similar meanings and the same case markers are given in Table 2.7 for languages Turkish and HuI came **from Ági**.ngarian, which have rich case marking systems.

In the middle column of Table 2.8, English sentences with different syntactic realizations and their translation into Turkish, and in the last column the role that wants to be emphasized in the sentences are given. In the first three sentences, all words written in bold represent the arguments in destination roles. English sentences can not describe a common syntax for the destination role; different prepositions such as "into", "at", "onto" precedes the argument. However, in Turkish sentences it is always marked with dative case. Similarly, in the last three rows of Table 2.8, source and a similar role initial location (IniLoc) are emphasized. Again, it is hard to find a distinguishing feature that reveals source and initial location roles in English sentences. There may be different prepositions "out of", "from" or no preposition at all, before the argument in one of these roles, but they are naturally marked with ablative case in Turkish sentences. These suggest that case markers can be a distinguishing feature for argument identification and classification steps of SRL.

Lang	Destination	Source
#1.En	She$_{\text{Ag}}$ loaded boxes$_{\text{Th}}$ into the **wagon**$_{\text{Dest}}$.	He$_{\text{Ag}}$ backed out of the **trip**$_{\text{Sou}}$.
#1.Tr	Kutuları$_{\text{Th}}$ **vagon-a**$_{\text{Dest-DAT}}$ yükledi.	**Seyahat-ten**$_{\text{Sou-ABL}}$ vazgeçti.
#2.En	She$_{\text{Ag}}$ squirted water$_{\text{Th}}$ at **me**$_{\text{Dest}}$.	The convict$_{\text{Ag}}$ escaped **the prison**$_{\text{iniLoc}}$.
#2.Tr	**Ban-a**$_{\text{Dest-DAT}}$ su$_{\text{Th}}$ fışkırttı.	Mahkum$_{\text{Ag}}$ **hapis-ten**$_{\text{iniLoc-ABL}}$ kaçtı.
#3.En	Paint$_{\text{Th}}$ sprayed onto the **wall**$_{\text{Dest}}$.	He$_{\text{Ag}}$ came from **France**$_{\text{iniLoc}}$.
#3.Tr	**Duvar-a**$_{\text{Dest-DAT}}$ boya$_{\text{Th}}$ püskürtüldü.	**Fransa'dan**$_{\text{iniLoc-ABL}}$ geldi.

Table 2.8 : Relation between case markers and semantic roles. Ag: agent, Th: theme, Dest: destination, Sou: source, Pat: Patient

Lexically poor languages may suffer from homonyms. In such languages, one verb may be used to express many different things. The task of finding the meaning of word in the context in question is called word sense disambiguation. In Table 2.9 three senses of Turkish verb lemma "ayır" and their arguments with case markers are given. In the first sense, the arguments are marked with ACC and DAT, with ABL and NOM in the second and with ACC, ABL in the third. It suggests that case marker information can also be beneficial for predicate sense disambiguation step of SRL.

2.3.2 Derivational Morphology

Consider the word "belgelendirilememesindeki" [1] (*at the time of (it) not being able to get documented*) shown below. Here, although the final form of the word is an

[1] Example taken from [46]

	ayır.01 - To divide, split into pieces
#1.En #1.Tr	[He/she]$_{\text{Ag}}$ divided [the apple]$_{\text{Pat}}$ [into four]$_{\text{Dest}}$. [Elmay-**ı**]$_{\text{Pat-ACC}}$ [dörd-**e**]$_{\text{Dest-DAT}}$ **ayır**dı.
	ayır.02 - To keep, reserve (get-13.5.1)
#2.En #2.Tr	[I]$_{\text{Ag}}$ reserved [a table]$_{\text{Th}}$ [from the restaurant]$_{\text{Sou}}$. [Restoran-**dan**]$_{\text{Sou-ABL}}$ [masa]$_{\text{Th-NOM}}$ **ayır**dım.
	ayır.03 - To seperate (separate-23.1)
#3.En #3.Tr	[I]$_{\text{Ag}}$ separated [the yolk]$_{\text{Pat1}}$ [from the white]$_{\text{Pat2}}$. [Sarısın-**ı**]$_{\text{Pat1-ACC}}$ [beyazın-**dan**]$_{\text{Pat2-ABL}}$ **ayır**dım.

Table 2.9 : Relation between case markers and word senses

adjective, it undergoes five derivational changes shown with curly brackets. Noun root “belge" (document) derives into three different verbs, then noun and finally an adjective. The shift in the meaning and the type of the word is called derivation, and the morpheme that causes that shift is named derivational morphemes.

$$\underbrace{belge}_{Noun}+len+\underbrace{dir}_{Verb}+\underbrace{il}_{Verb}+\underbrace{eme}_{Verb}+\underbrace{me}_{Noun}+si+\underbrace{nde}_{Ad jective}+ki$$

Turkish is among languages with rich derivational morphology. Complex derivational morphology of Turkish poses two challenges to semantic role labeling. First, large numbers of productive derivational morphemes cause a theoretically **infinite word lexicon**. Secondly, derivational morphemes introduce wide range of syntactic variations for predicate-argument structures, the different ways in which the predicate and its argument structure can be realized. In order to illustrate these challenges, we investigate the treebank for frequently encountered derivations that involve verbs. In Table 2.10, such derivations together with their frequencies and example sentences are shown. For the sake of simplicity, we divide them into 3 categories according to their derivation types. In the first and the biggest category, predicates are observed as verbs (either in root or derived forms via valency changing morphemes). The second category represents verbs that are transformed into nouns, adjectives or adverbs, named verbal nominals. Finally, category 3 contains verbs that are derived from nouns, which we refer to as nominal verbs.

CATEGORY 1: NO DERIVATION or VALENCY CHANGE			
Row	Form	Count	Examples
1	ROOT	5277	Gösteri **başla**-malı. show **start**-Neces The show should **start**.
2	PASS	453	Kapı **vur-ul**-du. door **knock-Pass** The door **was knock**ed.
3	CAUS	235	Kolları-nı **birleş-tir**-di. arms-his **unite-Caus**-Past He **folded** his arms.
4	CAUS + PASS	44	Fiyatlar **düş-ür-ül**-dü. prices **go down-Caus-Pass**-Past The prices **have been reduced**.
CATEGORY 2: VERBAL NOMINALS			
Row	Form	Count	Examples
5	ROOT → Noun	1744	Kız **gel-eceğ-i-ni** söyledi. girl **come-Fut-she-Acc** said. The girl said **that she will come**.
6	ROOT → Adj	1320	**Yüz-en** biri var. **swim-Adj** someone be. There is someone **swimming**.
7	PASS → Adj	309	**İnan-ıl-maz** zeki biri. **believe-Pass-Neg** smart one An **unbelievably** smart one.
8	PASS → Noun	215	**Kaldır-ıl-ma**-sı gerek. **remove-Pass-Inf**-it necessary It needs **to be removed**.
9	PASS → Adv	60	**Sor-ul-unca** konuş. **ask-Pass-When** speak. Speak **when you are asked**.
10	CAUS → Noun	108	Ağzın-dan **kaç-ır-mak** tehlikeli. mouth-from **flee-Caus-Inf** dangerous. **To blurt out** is dangerous.
11	CAUS → Adj	95	**Rahatla-t-ıcı** şeyler söyle. **relax-Caus-Adj** things say. Say something **comforting**.
12	CAUS → Adv	42	Herkes-i **gül-dür-ünce** rahatla-dı. everyone-Acc **laugh-Caus-When** relax-Past He has relaxed **after making everyone laugh**.
13	CAUS + PASS → Noun	33	**Bekle-t-il-eceğ-ini** öğren-di. **wait-Caus-Pass-Fut-Acc** learn-Past He learned **that they will make him wait**.
14	CAUS + PASS → Adj	31	**Boşal-t-ıl-an** köyler orada. **empty-Caus-Pass-Adj** villages there. The villages **that were emptied** are there.
15	RECIP →Noun	34	**Sev-iş-mek** iste-mez. **love-Recip-Inf** want-Pres He doesn't want **to make love**.
CATEGORY 3: NOMINAL VERBS			

Row	Form	Count	Examples
16	Noun→ROOT→Noun	20	**Buz-lan-ma** yok. **ice-Acquire-Inf** no. There is no **icing**.
17	Noun→ROOT	34	**Merak-lan**-ıyor-um. **curiosity-Acquire**-Pres-I I'm getting curious.
18	Noun →CAUS	11	Adımları-nı **hız-lan-dır**-dı. steps-his **speed-Acquire-Caus**-Past He quickened his steps.

Table 2.10 : Derivational morphology of verbs in IMST. *Count*: Number of occurences in the treebank; ROOT: Root Verb; PASS: Passive; CAUS: Causative; RECIP: Reciprocal; *Adj*: Adjective; *Adv*: Adverb; Morphemes used in examples: *Neces*: Necessity; *Fut*: Future Tense; *Acc*: Accusative marker ;*Neg*: Negation; *Inf*: Infinitive

The overall aim of building semantic resources is to abstract away from syntactic idiosyncrasies as much as possible, i.e, to assign similar framesets to sentences with similar meanings. For example, *the market grew fast*, *growth of the market was fast* and *fast growth in the market* are desired to have same set of semantic frames. Moreover, the frequencies given in Table 2.10 show that only 43% of predicates are observed as roots while the rest undergo derivational changes. In order to abstract farther away from syntax i.e., to assign same framesets to examples above and address the remaining 57%, we propose to use framesets of root verbs for semantic role annotation of derived words. For example *boşaltılan* (*discharged*) (14 th row of Table 2.10) will use the frameset of *boşal* (*to empty*), *büyüme* (*growth*) will use *büyü* (*to grow*) and *düşürül* (*to be made drop/fall*) will use *düş* (*to drop*). [2]

This approach is also helpful to address the *infinite word lexicon* problem, since **a lexicon of root verbs** is sufficient to accomplish SRL. For example, frameset of *gül (to laugh)* is sufficient to semantically analyze a sentence that contains all syntactic variations of *gül* e.g., *gül-dür (to make someone laugh)*, *gül-üş (to laugh together)*, *gül-dür-ün-ce (when sb makes sb laugh)*, *gül-en adam (smiling man)*. Furthermore a small lexicon decreases the complexity of SRL systems based on machine learning methods, since the majority of models train a separate classifier per lemma. Moreover, it enables us to boost the performance of supervised machine learning systems by

[2]Some exceptional derived verbs like *görüş, tanış* (*to meet*) that acquired new uses have their own framesets.

increasing training data for individual verbs. For example by labeling all syntactic variations of *gül (to laugh)* written above with the same predicate roleset e.g., *gül.01*, the training data for the classifier is increased greatly.

The differences between Turkish and *other languages* in framing process mainly arise from valency changes and derivational morphemes that derive nouns from verbs. Verbal nominals can be annotated using framesets of root verbs, hence is not discussed in this section.

2.3.2.1 Valency Changes

Valency of a verb specifies the number and type of arguments a verb can govern. These morphemes are generally regular and exist for many languages with rich morphology. Similar to other derivational morphemes in Turkish, they are productive and can be concatenated numerous times. This causes an increase in verb lexicon size. According to Table 2.10, reflexive and reciprocal verbs are not as common as causative ones, but may occur frequently in other forms e.g., reciprocal verbs transforming into nouns as in row 15 of Table 2.10.

Causative morpheme [3] introduces a new argument, namely *the causer* to verb's semantic frame which we label as *ArgA*. [4] It causes different inflectional changes for transitive and intransitive verbs discussed in our preliminary work [47].

giy (*to wear*)	**giy-dir (*to make sb. wear sth.*)**
[Kız]$_{\text{wearer}}$ [ceket-i-ni]$_{\text{clothing}}$ giy-di.	[Oğlan]$_{\text{causer}}$ [kız-a]$_{\text{wearer}}$ [ceket-i-ni]$_{\text{clothing}}$ giy-**dir**-di.
girl jacket-ACC wear-PAST	boy girl-DAT jacket-ACC wear-**CAUS**-PAST
[The girl]$_{\text{wearer}}$ wore [her jacket]$_{\text{clothing}}$.	[The boy]$_{\text{causer}}$ made [the girl]$_{\text{wearer}}$ wear [her jacket]$_{\text{clothing}}$.

(a)

	Kız	ceket-i-ni	giy-di	.
giy.01	A0:*wearer*	A1:*clothing*		

(b)

	Oğlan	kız-a	ceket-i-ni	giy-dir-di	.
giy.01	A-A:*causer*	A0:*wearer*	A1:*clothing*		

(c)

Figure 2.3 : Causative derivation of transitive verb *giy (to wear)*. (a) Case marking of arguments for root and causative (b) Labeling semantic roles of giy *(to wear)* (c) Labeling semantic roles of giy-dir *(to make sb. wear sth.)*

[3] Most commonly used causative morphemes in Turkish are *–r*, *-DIr*, *-D*. (*D* represents the letters *d* or *t*, *I* is used to denote *i* or *ı*).

[4] ArgA represents the only causer present or the external most causer in case of multiple causers.

In Fig.2.3 (a) derivational morpheme introduced *the boy* as the causer and *girl (the agent)* is marked with the dative case marker; while the clothing *jacket* does not undergo any lexical change. In Fig.2.3 (b) and (c) semantic role analysis for the root verb and the derived verb are shown. They are both annotated with the frameset *giy.01* and independent from syntactic variations of the arguments *girl* is consistently annotated as *the wearer* and *jacket* as *the clothing*.

Though framesets of causative transitive verbs are compatible with English PropBank, some causative intransitive verbs differ from English equivalents. Consider the sentences:

(1) [The eggs] **mix**-ed [with the cream]. (2) [Herman] **mix**-ed [the eggs].

(1) [Yumurtalar] **karış**-tı [kremay-la] (2) [Herman] **karış-tır**-dı [yumurtaları].

Here, the verb *mix* translates as an intransitive Turkish verb *karış* in the first sentence but as its causative derivation *karış-**tır*** in the second one. According to framesets for mix.01 and karış.01 given in Table 2.11, while Herman in sentence (2) would be labeled as *A0: agent, mixer* by English PropBank, he would be labeled as *A-A: causer of mixing event* by Turkish PropBank. This example with the verb *karış (to mix)* is not an exception and repeats for many intransitive Turkish verbs with causative forms such as *değiş (to change)*, *piş (to cook)*, *uç (to fly)*, *rahatla(to relax)*, *taşı (to move)* and more. Since Turkish and Finnish are from the same language family, Finnish PropBank [14] faces a similar issue. They create separate frames for the root verb and its causative derivation by introducing constraints in order to stay consistent with English PropBank framesets. On the contrary, we continue to use the root verb's frameset to keep our resource as invariant of syntax and self-consistent as possible.

mix.01	**karış.01**
A0: agent, mixer	_
A1: ingredient one	A1: karışan şey (mixing thing)
A2: ingredient two	A2: neyle karıştığı (with what)
A3: end product	_

Table 2.11 : Framesets for mix.01 and karış.01

Our approach of reusing the root verb's frameset with an additional causative role instead of creating a new frameset for the causative verb has many advantages. First of all, it helps us reduce the need for the most valuable resource: experts for creating new

framesets. Moreover, it enables us to abstract away from syntax *without introducing any additional constraints*. In example from Fig. 2.3, we annotate *girl* as *the wearer (Arg0)* and *jacket* as *the clothing* in both sentences, invariant of syntactic changes. If we had used an English compatible frameset for *wear.01* as Finnish PropBank, *girl* would be annotated as *person wearing clothes (Arg1)* in both sentences again invariant of syntactic changes. However, an additional constraint that forbids the predicate *wear.01* from governing an *Arg0* in the first sentence had to be introduced. Furthermore, having a smaller number of predicate lemmas helps the supervised machine learning method in terms of 1) providing more training data for each lemma and 2) reducing the necessary number of separate classifiers built for each lemma. Only disadvantage of our approach is the reduced compatibility to the English PropBank by means of semantic roles for causative verbs.

Reciprocal verbs serve the purpose of expressing actions that are done together or against each other. In Table 2.12, two sentences with reciprocal verb *döv-üş (to fight)* are given. When the agent is plural as in first example, the reciprocal morpheme suppresses one of the arguments and replaces the single agent *boy* with plural agent *men*. In the second case, co-agent *guardian* is marked with the commitative case marker.

döv (*to beat*)	**(a) döv-üş (*to fight*)**	**(b) döv-üş (*to fight*)**
[Oğlan]$_{\text{hitter}}$ [bekçi-yi]$_{\text{thing hit}}$ döv-dü.	[Adamlar] döv-**üş**-tü.	[Oğlan]$_{\text{hitter}}$ [bekçi-yle]$_{\text{thing hit}}$ döv-**üş**-tü.
boy guardian-ACC beat-PAST	men beat-**RECIP**-PAST	boy guardian-COM beat-**RECIP**-PAST
[Boy]$_{\text{hitter}}$ beat [the guardian]$_{\text{thing hit}}$	[Men] fought.	[Boy]$_{\text{hitter}}$ fought with [the guardian]$_{\text{thing hit}}$

Table 2.12 : Example sentences for reciprocal verb *döv-üş (fight)* with (a) plural agent and (b) co-agent

Again, we use the frameset of the root *döv (to beat)* for derived verb *döv-üş (to fight)* for consistency among annotations e.g., *bekçi (the guardian)* is labeled with *A1: person hit* for both sentences in Table 2.12. As a side note, English PropBank does not distinguish plural agents from single ones. We follow these guidelines and label the plural agent *adamlar (the men)* in Table 2.12 (b) with *A0: hitter* semantic role.

Reflexive verbs are used to define the action that directly affects the person or the thing who does the action [48]. The reflexive morpheme associates the agent *hider* with the patient *thing hidden* and squeezes them into one argument as shown in Fig. 2.4. Regarding annotation of reflexive verbs, it would not be wrong to annotate *kız (the*

sakla (*to hide*)	**sakla-n (*to hide himself/herself*)**
[Kız-ı]$_{\text{thing hidden}}$ sakla-dı-lar.	[Kız]$_{\text{hider\&thing hidden}}$ sakla-**n**-dı.
girl-ACC hide-PAST	girl hide-**REFL**-PAST
[They]$_{\text{hider}}$ hid [the girl]$_{\text{thing hidden}}$.	[The girl]$_{\text{hider}}$ hid [herself]$_{\text{thing hidden}}$.

Figure 2.4 : Argument suppression example for reflexive verb *sakla-n (to hide himself/herself)*

girl) in Fig. 2.4 as *A0: hider* and *A1: the thing hidden*. However PropBank conventions do not allow us to annotate one argument with two different roles. In this particular example, *being hidden* seems more important than *hiding something*, therefore *A1* may be a more appropriate label for *the girl*. However in most cases, deciding for the dominant role is really hard because *A0* and *A1* seem to be equally important. In such cases, we prioritize *A0* over *A1* according to English PropBank guidelines that prioritize low numbered arguments over high numbered ones.

As also reported by Finnish PropBank, it is one of the most confusing scenarios for annotators. We believe it would be more convenient to define a new semantic role like A0A1 to account for multi-role arguments. Since frequency of multi-role arguments is low in the treebank, they are left as a future work. Furthermore, English and Hindi PropBanks use AM-REC (Reciprocal) to label reflexive pronouns e.g., himself/herself which are generally implicit in Turkish.

2.3.2.2 Nominal Verbs

We use the term nominal verbs to identify verbs derived from nominals as shown in CATEGORY 3 of Table. 2.10. Turkish morphemes that derive verbs from nouns such as *-lA, -lAş, -lAn* can transform vast amount of Turkish nouns, and therefore are important for a high coverage PropBank. For example the morpheme *-lAş* produces the verbs *sert-leş (to harden)*, *sessiz-leş (to become quiet)*, *sevimli-leş (to become adorable)* while the morpheme *-lAn* produces *heyecan-lan (to get excited)*, *hüzün-len(to get sad)* , *pahalı-lan (to become more expensive)*. In order to address the issue of nominal verbs which was ignored in our preliminary work [49], we create frames for morphemes as *xlA, xlAş, xlAn* where “x" represents the noun root.

2.4 Lexicon Statistics

Table 2.13 shows that 583 out of 773 lemmas (approximately 75% of the Turkish PB lexicon) have only one sense/roleset and 1,66 senses have been created per lemma on average. Table 2.14 states that rolesets are likely to have two roles (58% of rolesets),

#sense	**#lemma**	**#sense**	**#lemma**
1	583	**8**	4
2	113	**9**	2
3	40	**11**	1
4	10	**12**	1
5	6	**13**	2
6	4	**18**	2
7	2	**26**	2
61	1		
Total:	#senses: 1285		#lemmas: 773
Average:	1,66		

Table 2.13 : #sense: Number of senses; #lemma: Number of lemmas

while the average number of roles per sense is 2,11.

1	**2**	**3**	**4**	**5**	**6**
210	750	303	17	4	1
Total:		#roles: 2713		#senses: 1285	
Average:	2,11				

Table 2.14 : Columns: number of roles, Rows: number of senses

2.5 Summary

We illustrated immense syntactic variation and infinite word lexicon problem caused by derivational morphemes and proposed exploiting framesets of root verbs to address them [47,49]. We discussed how this approach enables us to abstract farther away from syntax and increase self-consistency of Turkish PropBank. We presented the general procedure of the framing process with guidelines for sense and argument number distinctions. We explored issues raised by valency changing morphemes, light verbs, multiword expressions and nominal verbs; and demonstrated how case markers can be beneficial for SRL task. In conclusion, this chapter presented a lexicon of Turkish verbs consisting of 773 lemmas and 1285 senses that are framed with PropBank annotation scheme.

3. Annotation of Turkish

This chapter uses the outputs of the Framing chapter (Chapter 2) to create a semantically annotated corpus. First of all, we introduce the IMST corpus which we perform the verb sense and semantic role annotations on. Afterwards, we demonstrate the feasibility of our approach on derivationally complex real-world sentences taken from the corpus. Then, crowdsourcing of verb sense disambiguation and semantic role annotation (see Fig. 1.4) are represented. We discuss the quality control mechanisms that we have used and present evaluation results for overall annotation quality of the resource.

3.1 Background and Related Work

Crowdsourcing approaches have recently been harnessed to annotate, evaluate and create corpora [50, 51] for different NLP problems such as sentiment analysis [52], machine translation [53], grammatical error detection [54], named entity recognition [55] and word sense disambiguation [56, 57]. Evaluation of crowd annotations for NLP tasks by Snow et.al [58] has shown that the resultant data is high quality and enhances NLP systems. Reported results have encouraged some tutorials [59] and best practice guidelines [60]. Despite high employment rates of crowdsourcing platforms for NLP tasks, it has not been utilized by complex semantic tasks such as **Semantic Role Annotation (SRA)** until recently. Fortunately, increasing interest on semantics provoked the community to exploit crowdsourcing on semantic annotation tasks.

Feizabadi et.al [61], created two different HITs (Human Intelligence Tasks) to annotate implicit frame-semantic roles (Place, Source, Goal, Path) and present new resource with an acceptable quality. Frame disambiguation and element identification are performed in a single step [62] and later improved by leveraging information from DBpedia [63]. Frame disambiguation is performed while emphasizing the importance of exemplars and real-time feedback [64]. Reisinger et.al developed [65] an annotation

task based on the idea that thematic roles should be decomposed into fine-grained properties and tested it on real world data from PropBank corpus.

One of the biggest challenges of SRA task is its complexity for the crowd without linguistic expertise. In order to overcome this problem, He et.al [66] generate questions that correspond to semantic roles *e.g., Who beat someone?* for the predicate *beat*. Instead of using an ontology for predicate senses and semantic roles, they treat each verb sense equally and focus on relating questions to semantic roles. To simplify the SRA task, Rim et.al. [67] supply an annotated example with the predicate and one of its arguments with a specific semantic role (e.g., *sell* and *sellee*) and asks the taskers if the argument in the second sentence have the same role. Although semantic annotation is considered complex for non-experts, all of the studies above have reported good quality data and high inter-annotator agreement between the crowd and gold data.

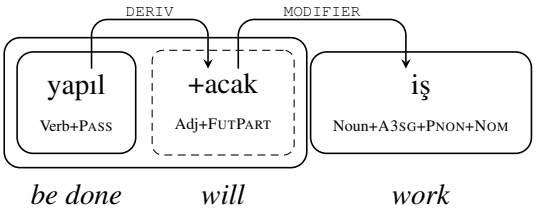
Due to demographics of crowdsourcing platforms, most of the studies mentioned above have been applied to one of the 13 languages that were reported as good candidates to work with [68]. The languages outside this list such as Turkish have generally been investigated in a bilingual machine translation (MT) evaluation framework [69]. Moreover, most of the previous works on crowdsourcing SRA use FrameNet scheme and focus on a small number of predicates and semantic roles to simplify the task. Contrary to previous works, we scale up the task to annotate 20.000 semantic roles from 20 categories for a language considered low-resourceful.

3.2 Corpus

IMST is a syntactically annotated corpus with sentences from Metu Turkish Corpus [70] [1]. It contains modern Turkish text from 10 distinct genres. It has over 56.000 orthographic words (word's surface form) and 63.000 syntactic tokens. It is morphologically analysed, POS tagged, and annotated with dependencies based on the grammar defined by its ancestor Metu Sabancı Treebank (MST) [71] with minor revisions. Briefly, words are represented as a sequence of inflectional groups (IGs) separated by derivational boundaries (DBs), where IGs include inflectional features and POS information and dependency relations are between IGs rather than

[1] http://www.ii.metu.edu.tr/~corpus/corpus.html

orthographic words. For demonstration purposes, the phrase “yapılacak iş" (*work that will be done*) is represented as:



Here, the first IG of *yapılacak (that will be done)* has the *Verb* POS tag which is later inflected with passive voice. The second IG indicates a derivation into an adjective to form the word *yapılacak*. These IGs are linked with a special dependency label DERIV that denotes a derivational boundary (DB) and dependency links always emanate from the last IG of a word as in the example. Furthermore, IMST adds *deep dependency* annotations for multiple headed arguments to allow identification of indirect arguments of predicates. IMST is later mapped to universal dependencies (UD) and released as IMST-UD.

Unlike many other dependency treebanks, IMST provides links that enables nodes to have multiple heads as shown in Fig. 3.1. These links are called *deep dependencies* and enable identification of indirect arguments of predicates. Sulubacak et.al [70] report that 3,8% (which is equal to 2.401 links) of all dependency links are deep in IMST.

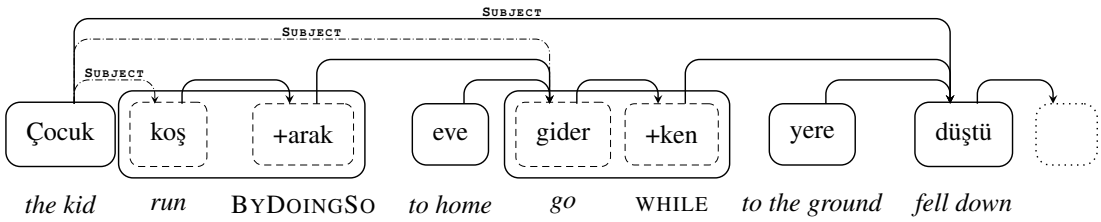


Figure 3.1 : The argument *the kid* is shared among the predicates *run*, *go* and *fall down* in the sentence *The kid fell down while going home by running*. SUBJECT links shown with dotted lines refers to deep dependency links.

Similar to *deep dependency* annotations, Finnish PropBank introduces *additional dependencies* layer and states the clear benefit of the layer for PropBank annotation [14]. Likewise, Stanford dependencies are *enhanced* with additional and augmented relations and later extended for UD representation [72]. Although their names differ, they all aim to capture otherwise implicit links between tokens by

transforming tree structures into graph structures and move syntax closer to semantics to provide improved representations for high level natural language tasks.

3.3 Feasibility

In this section, we discuss how our approach of using framesets of root verbs for semantic role annotation of derived words, copes with categories of Table 2.10. The annotation of nominals derived from verbs i.e., verbal nominals and their relation to the Nominal Bank is examined in 3.3.1. Finally verbs derived from nouns i.e., nominal verbs and our “framesets for morphemes" approach is described in Section 3.3.2.

3.3.1 Verbal Nominals

Annotation of verbal nominals is a part of semantic role annotation process and is also closely related to the project of Nominal Bank (NomBank) [20] which creates PropBank like frames for nominals. The high number of verbal nominals in IMST (around 39%) and inclusion of nominal predicates into SRL related shared tasks, make them an issue of great importance. Although NomBank takes all nominals

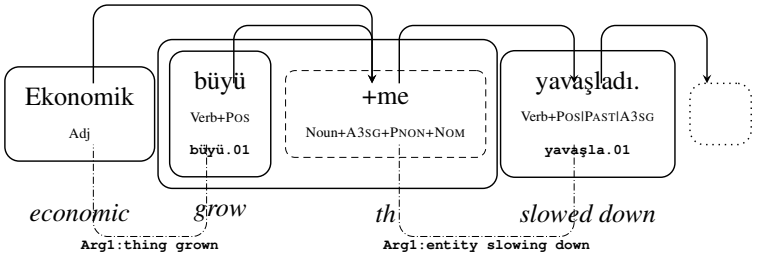


Figure 3.2 : Semantic association of *ekonomik (economic)* to verb *büyü (to grow)* instead of noun *büyüme(growth)*.

into account, we only consider nominals that are derived from verbs such as *(e.g., growth- (to grow)*, *tendency-(to tend)*, *thought-(to think)*, *discussion-(to discuss)* and *believer-(to believe)* for practical purposes. NomBank exploits framesets of relative verbs, but the link is indirect for SRL systems (generally different classifiers are trained for verbal and nominal predicates). However Turkish derivational morphemes provide direct links between verb and verbal nominal and therefore eliminate the need for creating nominal frames and feed the classifier for verbal predicate lemma with more training data.

As can be seen from CATEGORY 2 of Table 2.10, syntactic variation is very high among verbal nominals. In Fig. 3.2, the semantic role annotation for the noun *büyü+me (growth)* is illustrated [2]. *büyü+me* has a derivational boundary that seperates the word into two IGs: the verb *büyü (to grow)* and the noun *büyü+me (growth)*. We annotate the IG containing the verb with *büyü.01 (grow.01)*, so that *ekonomik (economic)* can be labeled with *Arg1:thing grown* role from its frameset. Although *ekonomik (economic)* is not directly linked with the verb, we can still identify it as an argument candidate by means of special DERIV labels.

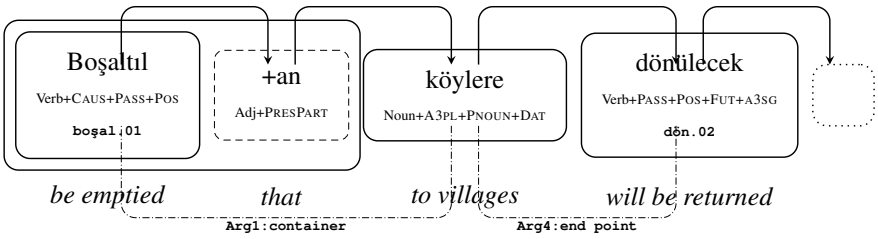


Figure 3.3 : Semantic annotation of adjective *boşaltılan (discharged)* derived from *boşal (to empty)*.

Adjectives derived from verbs are very common in Turkish and generally function similar to participles and relative clauses in English. In Fig. 3.3, the annotation of adjective *boşaltılan (discharged)* derived from verb *boşal (to empty)* is demonstrated. Following the morphological analysis of the predicate, it can be observed that first the valency is changed with the causative morpheme *t*, then verb is passivized with *ıl* and finally derived into an adjective with the present participle (PRESPART) tag. Later the adjective modifies the noun *köyler (villages)*. Similar to Fig. 3.2, the IG of the verb is marked with the roleset of the root *boşal.01 (empty.01)* and the modified noun is labeled with *Arg1:container* role. Unlike Fig. 3.2, the argument is observed as HEAD of the predicate in Fig. 3.3. A relatively more complex example for the semantic role annotation of the adverb *güldürünce (when sb make sb laugh)* derived from the causative verb *güldür (to make sb laugh)* is given in Fig. 3.4.

Fig 3.5 demonstrates an exceptional case where the argument is implicitly stated in a suffix. Here, *the things* is implicit in the noun IG *ları*. Object pronoun suffixes may be ambiguous, albeit not in this example, and thus are omitted in this study. Similarly

[2] In Figures 3.2,3.3,3.4, 3.5, 3.6 and 3.7 dashed boxes represent IGs containing derivational morphemes. Morphological analysis and rolesets of predicates are given *(when available)* at the bottom of the box. Top and bottom arcs depict syntactic dependency and semantic relations respectively.

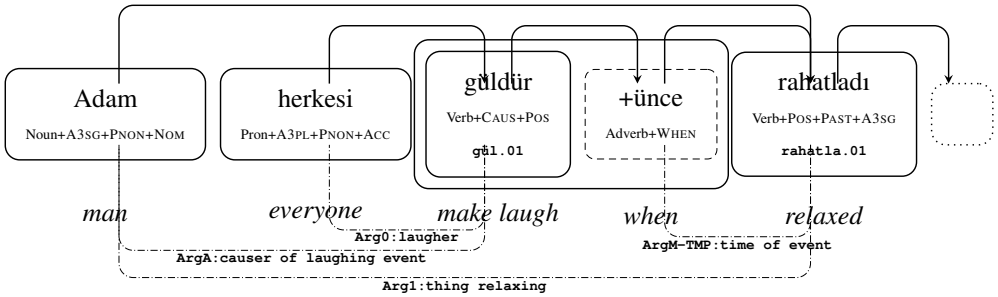


Figure 3.4 : Analysis of sentence *Man relaxed after making everyone laugh* with derived adverb *güldürünce (when sb makes sb laugh). Adam (man)* annotated with *ArgA:causer*, *herkes (everyone)* with *Arg0:laugher* defined by *gül.01 (laugh.01). güldürünce* is labeled with *ArgM-TMP: temporal adjunct Adam (man)* with *Arg1: thing relaxing* for the verb *rahatla.01 (relax.01)*

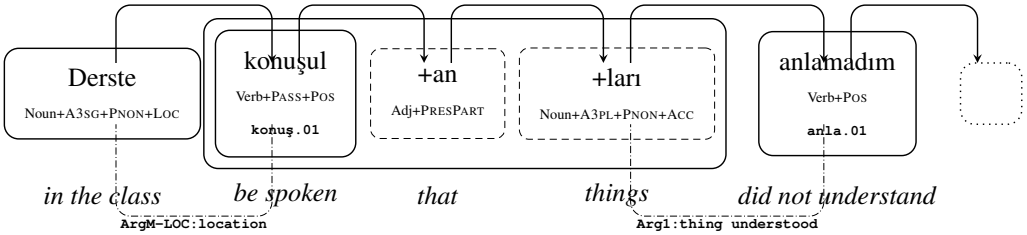


Figure 3.5 : Semantic structure of sentence *I didn't understand the things spoken in the class. Konuş (to speak)* first transforms into adjective, then into noun. *Derste* is labeled as locative argument of *konuş.01*; derived noun *konuşulanları* (*things that are spoken*) as Arg1 of *anla.01 (understand.01)*

we avoid labeling person suffixes attached to verbs, since they can directly be inferred from morphological tags and annotation of them would only create an overhead.

Regardless of derivational complexity, by using the relevant root verb's frameset, verbal nominals are annotated with semantic roles. Moreover syntactic transformations by these morphemes are generally unambiguous and can be learned via available machine learning algorithms. Although nominals discussed in this section would only make a small portion of a potential Turkish NomBank, a subset can be addressed even in absence of a Turkish NomBank.

3.3.2 Nominal Verbs

In Fig. 3.6, semantic analysis with frameset *xlAn.01* is given. Although it is the frameset of a morpheme, its argument structure resembles frameset of *get.03 (become)* and *become.01 (change of state)*.

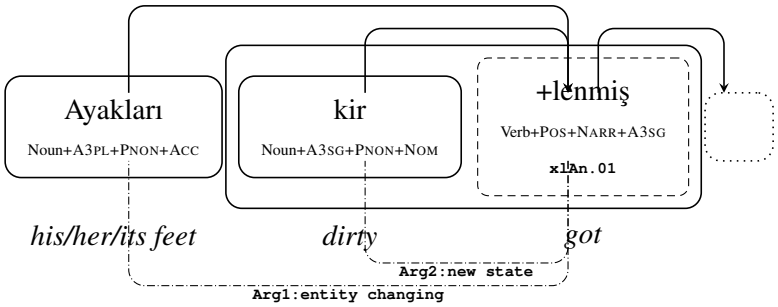


Figure 3.6 : Semantic arguments of *morpheme* frameset “xlAn.01".

3.3.3 Copula

Copulas are generally encountered in form of a morpheme attached to nominals (similar to nominal verbs discussed in Sec. 3.3.2) albeit with a few exceptions. Copulas may be of different types such as zero, *to be*, negative, personal, past and conditional [73]. Copular constructions in Turkish have many different types and deserves another dedicated study. We avoided to discuss them under *nominal verb* section because not all copular constructions are considered derivations. In some cases such as complex time structures (e.g., was going to), copulas can also be observed as inflectional morphemes. An annotated example with the most common copula type *to be* is given in Fig. 3.7. For this copula we have created a new roleset similar to *(be.01)* with two arguments, one for the thing that is, and other for what the first argument is, then labeled copulas for these semantic roles.

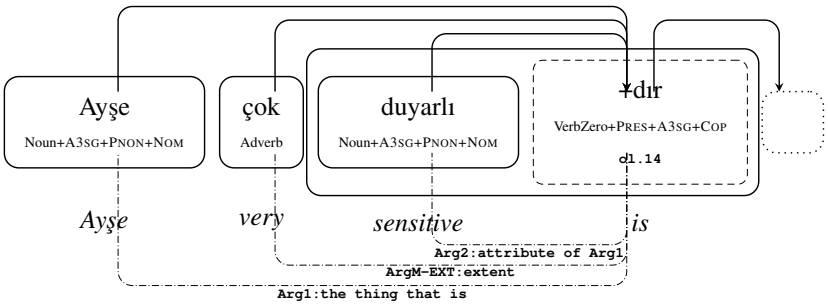


Figure 3.7 : Semantic roles of the copula “ol" *(to be)*.

3.4 Crowdsourcing Linguistic Annotation

Although semantic annotation is considered complex for non-experts, all of the studies above have reported good quality data and high inter-annotator agreement between the crowd and gold data which encouraged us to employ crowd taskers for VSD and SRA

tasks. Most common platforms of human intelligence tasks are Amazon Mechanical Turk [3] and CrowdFlower [4]. CrowdFlower's support for low-resourceful languages such as Turkish, ease of use due to its mark up language with conditional logic ability. Large number of natives from wide variety of cultures and improved quality control system has made it the right platform for us to perform Turkish verb sense annotation task.

3.4.1 Verb Sense Disambiguation (VSD) Task

VSD consists of disambiguating the meaning of the verbs that are encountered in the treebank. The *meaning of the verbs* correspond to the rolesets of predicate lemma, i.e., rolesets available in verb's frameset. First of all we have identified all words in the treebank that have an IG (Inflectional Group) with the Verb POS tag. There were total amount of $\sim$12.000 predicates that need to be annotated with corresponding rolesets. In other words all kind of derivations (Root Verb, Verb$\rightarrow$Verb, Verb$\rightarrow$Nominal, Nominal$\rightarrow$Verb) are included in this study. In order to reduce costs of the verb sense disambiguation task following preprocessing steps have been performed before the corpus has been outsourced to crowd:

- The predicates with only one sense have been eliminated,
- The predicate *ol (to be/become/happen)* has been eliminated since it is mostly used as light verb or copula, hence can be semi-automatically annotated with the help of dependency type to its head.[5]

A sample question shown to the taskers is given as follows.

[3] https://www.mturk.com

[4] https://crowdflower.com

[5] In Light Verb Constructions (LVC) with the verb *ol*, nominal dependent is linked with *MWE* dependency type to the predicate *ol*

Eylem: bit - 1
Predicate: *to finish - 1*
Tümce: Masal da burada bitmiş.
Sentence: *So the fairy tale ended here.*
Lütfen en yakın anlamı seçiniz:
Please choose the closest meaning:
(a)"Tükenmek, son bulmak" (Proje bitti.)
(a)"To end" (The project has finished.)
(b)"Çok sevmek" (Ben böyle sese biterim.)
(b)"To love so much" (I adore that voice.)
(c)"Hiçbiri"
(c)"None"

Here, first the taskers are given the predicate of interest along with its order in case of co-occurrences. Afterwards the context *i.e the sentence* of the lemma is given. Finally the contributors are supplied with descriptions of each roleset from Turkish PropBank. Later they are asked to choose the roleset that has the closest meaning to the predicate. They are also provided with *None* option in case of erroneous questions.

3.4.1.1 VSD Input Design for CrowdFlower

The design of input rows is given in Table 3.1. Here, only the first two senses of the predicate is shown however verb sense annotation task includes all senses that exist in Turkish PropBank. Each sense is included in a separate column as SENSEWEXi, where $i \leq 15$. In addition to the columns given in Table 3.1, additional attributes such

Field	Example
SID (Sentence Id)	3
SENT (Sentence)	Ona herşeyimi **verdim.** (En) I **gave him everything.**
PRED (Predicate)	ver (En) to give
PRNO (Predicate order)	2
SENSEWEX1 (First roleset)	Transfer etmek, iletmek (Yemlerini ben **veririm**.) (En) To transfer, transmit (I **give** them the bait.)
SENSEWEX2 (Second roleset)	Tespit etmek (İsim **vermek**, randevu **vermek**.) (En) To fix, establish (To **give** a name, to **fix** a date.)

Table 3.1 : Input row for Crowdsourcing Task

as unit id, state (finalized, judgable, golden), number of judgments done for this unit and agreement value between 0 and 1 are appended automatically by CrowdFlower.

Tasks with varying number of options such as verb sense annotation require dynamic rendering of questions. We have designed each sense in Turkish PropBank to be a new radio button. An example is shown in Fig. 3.8.

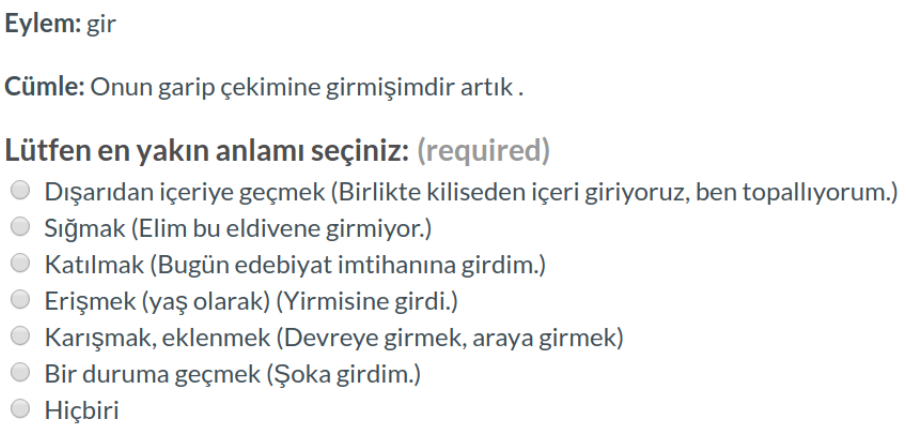


Figure 3.8 : Interface of verb sense disambiguation task

Since not all predicates have the same number of senses, we have dynamically rendered a new radio button when the column SENSEWEX*i* is not equal to "*N.A*".

3.4.2 Semantic Role Annotation (SRA) Task

SRA is the final and the largest step of the workflow and denotes assigning predicate specific semantic role labels to nodes in a dependency tree. As discussed previously, the list of expected arguments of each roleset is defined in the predicate's frame file and as referred to as **core** or **numbered** arguments and the set of general, adjunct-like arguments are named as *ArgM* or *AM*, where *M* is one of the function tags introduced in Table 2.5.

In order to keep the space of argument candidates as large as possible, we treat the graph as if it is undirected and identify all links i.e., *each head and dependent of each predicate* as an argument candidate. IMST contains around 20.000 argument candidates. All orthographic words that have a verb in their derivational process i.e., verbs derived from nominals (nominal verbs), nominals derived from verbs (verbal nominals) and verbs derived from verbs (valency change) are annotated with semantic roles. The annotated data has been prepared in the well known CoNLL-09 format and was later made compatible with the universal dependency format CoNLL-U.

We have designed the interface as follows: Given the sentence, predicate and the argument candidate, taskers are asked to decide the best semantic role for the potential argument. First, the contributor decides if it is actually an argument. Then he/she is shown the descriptions of core arguments extracted from the predicate lexicon of Turkish PropBank and asked to choose the best suiting semantic role. If she decides that none of them does, she is asked to choose one of the adjunct-like arguments from Table 2.5. We have automatically generated a “question unit" for each predicate-argument candidate pair encountered in our treebank. These question units are composed of fields extracted from the predicate’s frame file, the candidate argument, predicate’s order of appearance (necessary when the same predicate occurs multiple times in the same sentence), and the full sentence. (When a field is empty, it is assigned the value “N.A." (Not Available)). Once all fields are automatically filled with their corresponding values, questions are generated via the help of dynamic crowdflower markup language (CML). For the manual evaluation of the generated questions, we have added the *This question is incorrect* option at the end of each unit. Furthermore, we have put an optional text box to learn the reason why the taskers think the question is incorrect. One sample question shown to the taskers is given below. For convenience we only show three adjunct tags and omit the optional text box.

Tümce (*Sentence*): Ona her şeyimi **ver**dim. (*I **gave** her everything*.)
Eylem (*Predicate*): ver (*to give*)

Örnek 1: ***(Example 1:)***
Tümce: ***(Sentence:)*** Hayvanlara yemlerini ben veririm. *(I give the food to animals.)*
kime/neye verildiği: ***(entity given to:)*** Hayvanlara *(to animals)*
verilen şey: ***(thing given:)*** yemlerini *(their food)*
veren kişi: ***(giver:)*** ben *(I)*

Which defines the relationship of **“...Ona (*to her*)"** with the verb “ver (*to give*)" the best ?
(a)“İlişkili değil (*Not related*)"

One of the main relations:

(b)“veren kişi (*giver*)"

(c)“verilen şey (*thing given*)"

(d)“kime/neye verildiği (*entity given to*)"

If **not** any of the above:

(e)“Ettiren, yaptıran (*Someone/sth that makes/causes someone/sth to do sth*)"

(f)“Kiminle (Kardeşimle, NATOyla, onlarla) (*With whom ? (with my brother, with UN, with them)*)"

(g)“Nerede (okulda, konuşmasında, hayalinde) (*Where ? (at school, in his speech, in your dream)*)"

...(Other adjunct-like argument definitions)...

(x) This question is incorrect

Here, the predicate *ver* is already labeled as *ver.01* in the scope of the verb sense disambiguation task. As stated by Xue and Palmer [74], the frame files serve as lexical guidelines that ensure consistent annotation and provide a conceptual framework. To ensure that, we automatically extract annotated examples from *ver.01*'s frame file and substitute argument tags *(e.g., Arg0, Arg1)* with their descriptions *(e.g., giver, thing given)* to guide annotation and guarantee consistency. Furthermore at least one reference sentence annotated with predicate specific roles are shown right below the sentence with the label **Örnek *(Example)*** as shown above. Together with examples, the importance of clear/simple guidelines is emphasized by previous studies on crowdsourcing for semantic annotation [59, 64]. By following this advice, we provide simple but inclusive one line descriptions of adjunct-like arguments that are enriched with examples. For example PropBank annotation guidelines describe ARGM-EXT as *an indicator for the amount of change occurring from an action, and are used mostly for the following: 1. Numerical adjuncts, 2. Quantifiers such as a lot and 3. Comparatives*. However this definition is too long and may be hard to comprehend for taskers from different backgrounds. We simplify this description to **How much/many ? (50 %), (little, many), (more than him)** with small example phrases for each case. ArgM-LOC is similarly described as: **Where ? (at school, in his speech, in your dream)** as shown in the example question. Other than predicate

specific guidelines based on frame files, we provide more general, brief, step-by-step instructions including scenarios for complex cases formed by derivational morphology. These instructions are shown at the beginning of each page so that taskers can view them whenever they want to. Task design documents, instructions and full report of this crowdsourcing task can be downloaded from the project's website.

3.4.3 Configuration

We have configured both tasks to have three judgments per question, accept only native Turkish speakers and level 2 and 3 contributors as shown in Table 3.2. In addition to taskers from external channels, we have allowed internal team members to contribute to the task.

Setting	**Value**
Judgment Per Row	3
Rows Per Page	5
Payment Per Page	5 cents
Contributers' Level (1-3)	2
Channels	Internal and External
Geography/Langauge	Only Turkish
Minimum Confidence of Contributors	70%

Table 3.2 : Configuration of Verb Sense Annotation Task

3.4.4 Quality Control

Annotation quality is assured by following three procedures: Quiz mode before work mode, continuous training of taskers and removal of under-performers. Before contributors can start working on the actual task, they are requested to answer five test questions (expert labeled questions) and achieve at least the minimum confidence level. In other words, they are given X number of chances to fail before they move onto the work mode. X is calculated by *number of total questions in quiz mode* x *(1-minimum confidence level)*. For example if the number of questions in quiz mode is 10 and minimum confidence level is 0,80, then the annotators are allowed to fail maximum of 2 questions to start working on the task. This step is known as **Quiz mode** and helps us eliminate most of the under-performers from the beginning. For example in SRA task, only 36% of taskers could pass the quiz mode.

Continuous training of taskers is performed by guidelines and real-time feedback as also emphasized by [64]. Whenever a crowdworker fails a test question (whether in quiz or work mode), we provide the correct answer to test question and the explanation to make sure the mistake will not be repeated. At the bottom of the test question modification/monitoring window shown in Fig. 3.9, *Reason* text field can be seen. We have paid attention to fill those fields, so that the contributors can be informed whenever they miss any question. Finally, taskers that passed the quiz mode are

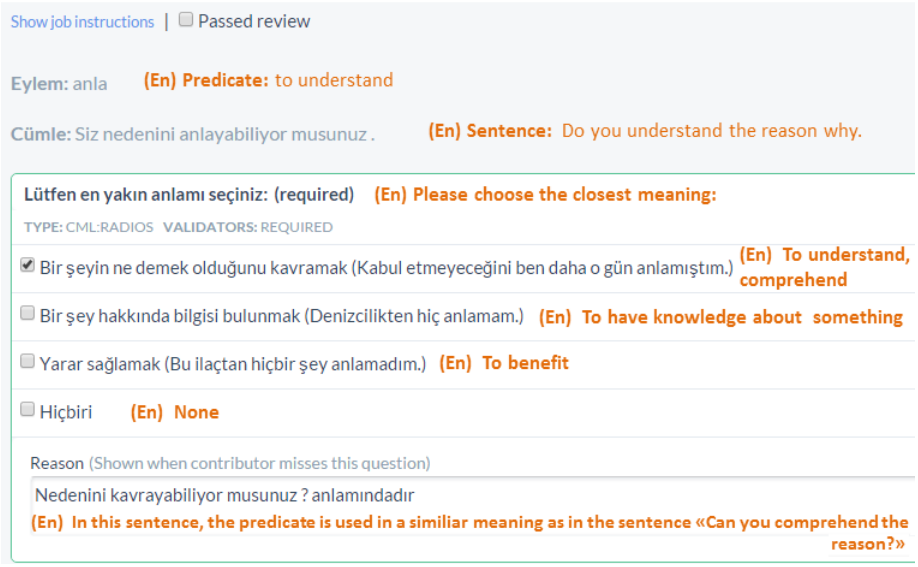


Figure 3.9 : Test Question Preparation View with *Reason* and *Passed Review* fields. Checkbox design allows marking of multiple answers for test question.

monitored during work mode. Each page contains one random test question and confidence levels of taskers are constantly updated as they answer them. When a crowdworker's performance drops below the minimum confidence level, he/she is immediately removed from the task. For instance 27% of the annotators that pass the quiz mode have been removed during work Mode during VSD task.

Test questions are the main components of our quality control mechanism. Therefore we have annotated around 2.000 (10% of all questions) golden units/test questions of varying difficulty. We have taken special care for having a variety of lemmas and rolesets in test questions and unambiguous answers for each question. Taskers are allowed to contest to expert answers if they think the provided answer is wrong. When the number of contests on a particular test question exceeds the threshold, CrowdFlower platform warns the task owner about that test question and asks it to be reviewed. As shown in Fig. 3.9, *Passed Review* button is used to indicate that the

contested question which has been reviewed by the expert. Some active contributors

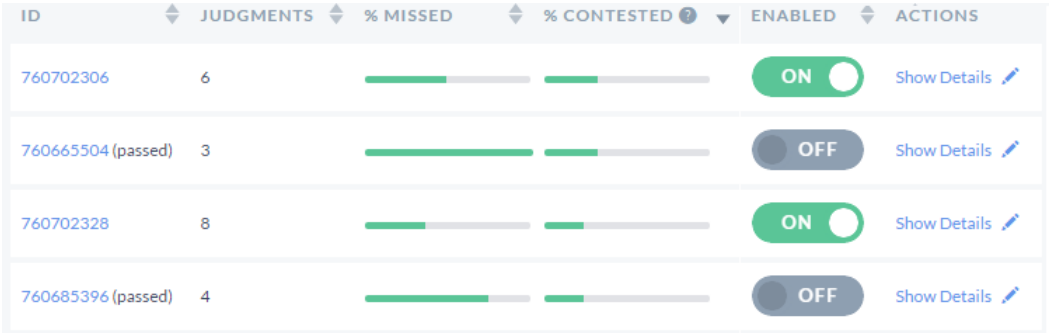


Figure 3.10 : Monitoring View of Test Questions: *Missed* and *Contested* indicates the ratio of missing/contesting the test question. *On/Off* Button can be used for including/excluding the test questions.

may memorize test questions during annotation process. Therefore, the task should be constantly monitored and test questions should be alternately enabled and disabled during annotation. In Fig. 3.10, test question monitoring and enable/disable buttons are shown. Another solution to prevent active contributors from memorizing test questions is to set maximum number of judgments per contributor roughly to be less than numberOfTestQuestions x 4 (Number of questions per page - 1). Maximum number of judgment per contributor is set to 10% of total rows by default.

3.5 Results

We present crowdsourcing results of VSD and SRA in the following subsections.

3.5.1 VSD Results

As a result, 5855 rows have been annotated and 18123 judgments have been made. 265,9 rows have been annotated per hour and all annotation process took 68 hours. More than 100 taskers contributed from 39 different cities of Turkey. The overall annotator agreement is calculated as 83,15% and the total cost of the job was 277 USD. The maximum amount of judgment made by one tasker is less than 800, which is only 4,44% of the job, as shown in Fig. 3.11. In Fig. 3.12, distribution of judgments with respect to the contributors' confidence is given. This figure shows that quality control mechanism of CrowdFlower eliminated the contributors with a confidence level lower than 70% which led to small amount of low-confident judgments. After completion of this task, the rows with confidence lower than 0,70 and ones that were agreed as *None*

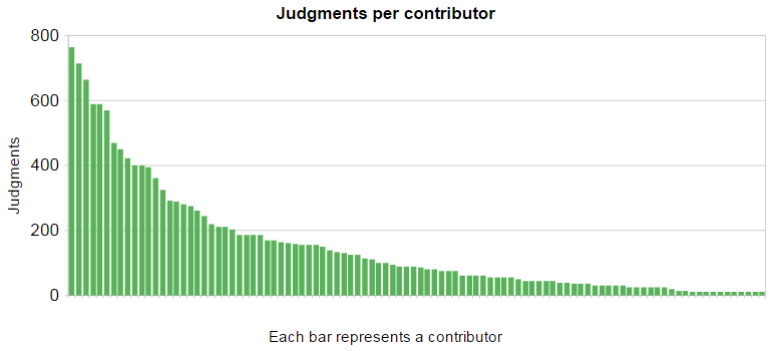


Figure 3.11 : Judgment per Contributor

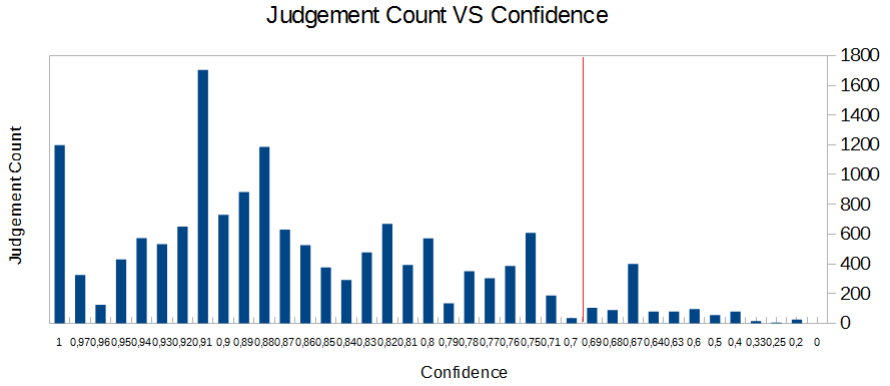


Figure 3.12 : Judgment vs Confidence. The vertical red line marks the confidence level 0,70

(when crowd taskers agreed that the right answer is not among options) have been revised. We have utilized this crowd feedback to improve the quality of verb frames. 2174 question rows had confidence lower than 0,70 and 738 rows were aggregated as *None* out of 6000 rows. We have manually performed a second pass annotation of experts for the rows with low confidence and eliminated 1200 out of 2174 of the rows since the aggregated results were already accurate. We have investigated the main reasons for annotators to choose the option *None* as follows and taken the appropriate actions;

- Mistakes in morphological analysis of the predicate such as analyzing the verb as *sok* (to put) instead of *sokul* (to get near); *kal* (to stay) instead of *kaldır* (to lift): These erroneous analyses have been corrected and the appropriate sense is chosen by an expert.

- Missing meanings: They are added to PropBank.

- Confusion caused by metaphorical expression: Verb senses are coarse-grained, thus metaphorical expressions are treated the same way as non-metaphorical expressions as suggested in PropBank guidelines [41].

Similarly, we have detected the causes of low-confidence rows as follows;

- Fine-grained verb senses: When two senses of the predicate have close meanings, it leads to confusions: Such frames were detected and merged.
- Missing meanings: They are added to PropBank.
- Confusing the meaning of the complete sentence with the meaning of the verb in question: They are revised and annotated by an expert.

As a result of this correction phase, the number of frame count increased from 675 to 759 and the total number of senses increased from 1135 to 1262.

3.5.2 SRA Results

A total of 20060 semantic roles have been annotated by more than 400 native Turkish speakers. Only 36,62% of them passed the quiz mode and among them 70% maintained the required confidence level. All annotations combined for a cost of 351 USD and took 276 hours.

As mentioned in Sec. 3.4.4, 10% of all questions were previously labeled by experts as test questions. First, we measure the annotation quality on these test questions in order to estimate the overall quality of the resource. We use two different measures: (1) inter-annotator agreement (2) precision, recall and F1. Inter-annotator agreement, in this case agreement between aggregated crowd answers [6] and expert labels, is measured via Cohen's kappa coefficient which is given as $\kappa = \frac{p_o - p_e}{1 - p_e}$. Here, p_o refers to relative observed agreement among annotators and, p_e refers to random hypothetical probability of agreement by chance. p_o, p_e and Cohen's κ are calculated as 0,948, 0,191 and **0,936** respectively. Precision (P) is given as the ratio of arguments labeled by the crowd that are also present in the gold standard, recall (R) is the proportion of arguments in the gold standard which are also present in the crowd output, and F1 is

[6] Aggregated answer is calculated as the agreement weighted by contributor trust by our crowdsourcing platform.

their harmonic mean. [7] P, R and F1 scores are calculated as 0,953, 0,956 and 0,954 accordingly as given in Test row of Table 3.3. These scores along with Cohen's κ coefficient indicate that annotations are of high quality.

Question Type	Precision	Recall	**F1**
Test	0,953	0,956	**0,954**
Other	0,902	0,903	**0,902**
All	0,906	0,908	**0,907**

Table 3.3 : Precision, Recall and F1 scores of crowd annotations. *Test*: Expert labeled 10%, *Other*: Unlabeled 90%, *All*: Combined 100%

However, test questions are generally chosen from a set of rows with *unambiguous* and *clear* answers in order to be fair with our taskers. For this reason, the afore-mentioned scores may only be an estimate for the overall quality of the resource. To get more insights on the annotation quality, we have calculated inter-annotator agreement among crowdworkers on questions without expert labels. Cohen's κ is designed to measure agreement between two annotators assuming the same raters have rated the same set of items. Therefore, we have used Fleiss κ coefficient which can handle more taskers and allows different items to be rated by different individuals like in our task [75]. It has been calculated as $0,65$, considering over 18.000 rows (all rows except test rows). Although there is no consensus on how to interpret Fleiss' κ scores, there is the fact that the score will be higher with fewer categories. Taking the large number of semantic role categories into account, $0,65$ is considered as substantial agreement.

In par with other PropBanks, we have observed that annotating numbered arguments was easier compared to temporary arguments. In order to show this phenomena we have calculated Fleiss κ for each semantic role category, shown in Table 3.4. We have seen that in addition to numbered arguments, agreements were high for temporal, locative, manner, extent, light verb and comitative roles.

Although crowdsourcing is reported to produce quality annotations, *adjudication* process is necessary to determine which semantic roles are aggregated correctly and should be included in the gold standard. Each semantic role is annotated by three taskers. After the crowdsourcing task, we have identified arguments where at least

[7] It should be noted that, we are able to calculate scores other than precision since annotators could decide if the argument candidate in the question is actually an argument. Therefore the set of arguments labeled by the crowd and the experts may differ.

Semantic Role	Fleiss κ	#	Semantic Role	Fleiss κ	#
Arg0	0,7649	3870	AM-MNR	0,5426	1486
Arg1	0,7453	7958	AM-EXT	0,5113	384
Arg2	0,7700	1354	AM-PRD	0,0763	144
Arg3	0,8065	291	AM-CAU	0,3257	359
Arg4	0,8835	625	AM-DIS	0,2039	138
A-A	0,3300	177	AM-ADV	0,2242	317
AM-COM	0,4022	84	AM-NEG	0,3272	67
AM-LOC	0,7089	856	AM-LVB	0,4417	694
AM-DIR	0,1861	78	AM-TMP	0,7562	1615
AM-GOL	0,3161	416	AM-INS	0,3034	167

Table 3.4 : Fleiss κ for each Semantic Role Category. # denotes number of occurrences

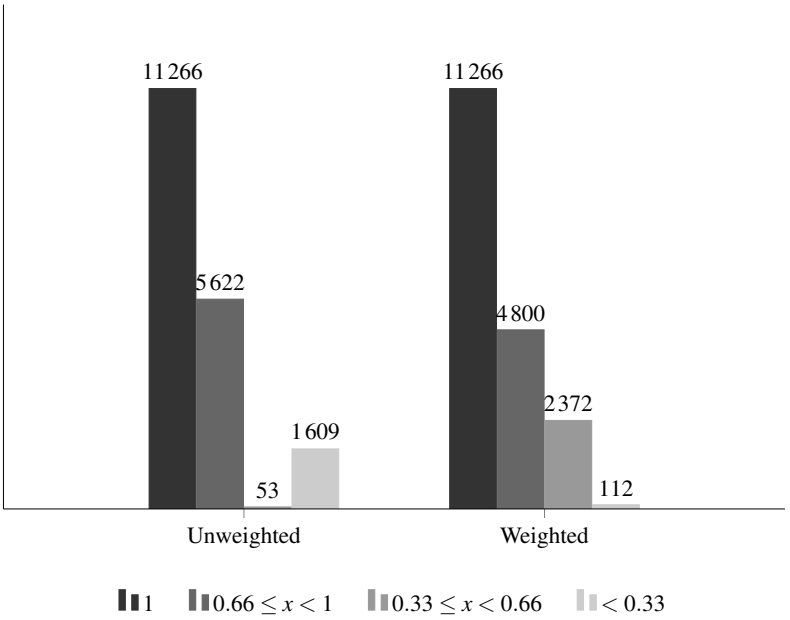


Figure 3.13 : Number of questions that fall into the four different agreement intervals

one annotation is different from other two answers i.e., arguments which do not have a full aggregation. These annotations are presented to an adjudicator, who is a more experienced annotator, to decide on the correct semantic role. There may be cases, albeit very rare, where the adjudicator determines all three annotations are inaccurate. In such cases, crowd answers are rejected and the adjudicator makes corrections.

We have calculated P, R and F1-score of crowd annotations on the PropBank after adjudication (separately for 90% of arguments (questions without expert annotation) and for entire PropBank including test questions) as provided in Table 3.3. F1 scores are slightly above 0,90 for unlabeled part of the PropBank and the entire PropBank, while above 0,95 for test questions as expected. In addition, we have

calculated agreement and weighted agreement on each question. An agreement on a question is calculated as (*maximum number of taskers that agree on the same label*)/(*total number of taskers that answer that question*), while a weighted agreement on a question is calculated as (*maximum confidence on the label*)/(*total confidence of taskers that answer the question*). For example, if the taskers with confidence 0,98 and 0,80 agree on a label and another crowdworker with confidence 0,70 chose another option, then the question agreement would be 0.66 and weighted agreement would be $(0,98+0,80)/(0,98+0,80+0,70) = 0,71$. Fig. 3.13 investigates agreement and weighted agreement in more depth. Unweighted refers to votes of all taskers are counted equal, Weighted is votes of taskers are weighted with their confidence level. Most test questions are answered by more then 3 taskers that led to the intermediate $0,33 \leq x < 0,66$ interval for unweighted scores. According to unweighted agreement intervals, 56% of the time taskers were in full agreement, and in 84% of the time at least two out of three crowdworkers agreed. However, when agreement scores are weighted with confidence level of the taskers, we observe a small drop in $0,66 \leq x < 1$ agreement interval, a large drop in $< 0,33$ agreement interval, and a big jump in $0,33 \leq x < 0,66$ interval. However on average, both agreements are measured as 0,84.

Furthermore, we have calculated confusions between the final (after adjudication) and **aggregated** crowd answers which are given in Table 3.5 and Table 3.6. Table 3.5 shows the confusion matrix for numbered arguments, where all adjunct-like arguments are represented with the label ArgM, where ArgM labels collapsed into one category. Entries are a fraction of total annotations; true zeros are omitted, while other entries are rounded to zero. According to the table the majority of the confusions were due to interchangeable temporary arguments with some core arguments. Even though it has been discouraged, some taskers chose temporary arguments over numbered ones (e.g., path and goal argument over Arg4). Another source of confusion was a framing mistake that overlooked important core roles for two frequent predicates. That was the major contributor to Arg1-ArgM and Arg0-ArgM confusions. Another difficult distinction was between Arg0 and Arg1 which was mostly triggered by multi-role arguments seen in reflexive valencies as discussed in Section 2.3.2.1. Table 3.6 is the confusion matrix for the secondary tags for ArgMs, which shows that even crowd taskers did not have a high degree of agreement for some ArgMs such as

	Arg0	*Arg1*	*Arg2*	*Arg3*	*Arg4*	*ArgM*
Arg0	0,155	0,010	0,000	0,000	0,000	0,007
Arg1		0,333	0,002	0,000	0,000	0,021
Arg2			0,059	0,000	-	0,005
Arg3				0,013	0,000	0,000
Arg4					0,030	0,017
ArgM						0,281

Table 3.5 : Confusion matrix for argument labels

DIS, the aggregated answer that was weighted by contributor trust was in line with the adjudicator's choice.

	A	ADV	CAU	COM	DIR	DIS	EXT	GOL	INS	LOC	LVB	MNR	NEG	PRD	TMP	TWO
A	0,004	0,000	0,000	-	0,000	-	-	0,000	0,000	0,000	-	0,000	-	-	0,000	-
ADV		0,007	0,000	-	0,001	-	0,000	0,000	-	0,000	-	0,000	0,000	0,000	-	0,000
CAU			0,010	-	0,001	-	0,001	-	-	0,000	0,000	0,001	0,000	0,000	-	0,000
COM				0,002		-	-	0,000	0,000	0,000	-	0,000	-	-	-	-
DIR					0,002	-	-	-	0,000	-	0,000	0,000	-	-	0,000	-
DIS						0,003	0,000	0,000	-	-	0,000	0,000	-	0,000	0,000	-
EXT							0,013	0,000	-	-	-	0,001	0,000	0,000	0,001	-
GOL								0,011	0,000	0,000	0,000	0,000	0,000	0,000	0,000	-
INS									0,005	-	0,000	0,000	0,000	-	-	-
LOC										0,029	0,000	0,000	0,000	0,000	0,000	-
LVB											0,028	0,000	-	0,000	-	0,000
MNR												0,059	0,000	0,000	0,002	0,000
NEG													0,002	-	0,000	-
PRD														0,002	0,001	0,000
TMP															0,067	-
TWO																0,003

Table 3.6 : Confusion matrix for secondary tags. Entries are a fraction of the total number of arguments, including core arguments

3.6 Post Annotation

Before we reach the gold standard Turkish PropBank, arguments that span over different parts of a sentence need to be addressed. Crowd workers annotate arguments individually, without considering the relation between arguments. For this reason, a predicate may have multiple arguments with the same role label which is forbidden by the nature of SRL. Such arguments are traditionally labeled as C-XXX for continuity and R-XXX for relative pronouns. To address them, we automatically identify predicates that have multiple arguments with the same semantic role and manually add the prefix C or R to their label. Fig. 3.14 illustrates annotation of a relative pronoun with the *R-* prefix. In this sample sentence, crowdworkers annotated *Sen ne dersen (Whatever you say)* and *o (it)* separately as the Arg1:event (i.e., thing that will happen) of *ol.01 (happen.01)*. Since *it* is the relative pronoun referring to *whatever you say* it has been labeled with R-Arg1. 0,010 of total arguments are labeled with prefix C-XXX while only 0,002 are annotated as R-XXX.

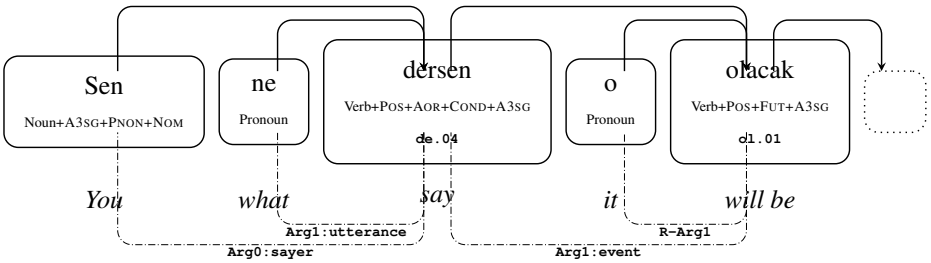


Figure 3.14 : Illustration of R-XXX in the sentence “Whatever you say, it will happen."

Out of all predicates in the corpus 0,056 of them are copulas. Annotation of copulas have been omitted in the scope of the crowdsourcing task, since their argument structures are misaligned with other verbs which may easily lead to confusion.

As part of the general trend to move from language specific annotation schemes to unified schemes across languages in the past few years helped UD project [8] to emerge. In scope of the UD project, IMST has been automatically mapped to the UD scheme and released with the name IMST-UD together with 40 other languages [76]. To support researchers working with the UD scheme, we have automatically aligned the

[8] http://universaldependencies.org/

semantic annotation layer of IMST, to IMST-UD and have released this UD compliant resource together with Turkish PropBank.

3.7 Discussion

Prior to launching the task to crowdworkers, we were concerned about the following issues:

1. No matter the derivational processes predicate goes through, taskers need to annotate with respect to the root verb's frame file. Consider the following example:

   > **Rahatla**tıcı şeyler söyle. (*Say things that cause me to* ***relax***.)
   > Which defines the relationship of "...şeyler (*things*)" with the verb "rahatla (*relax*)" the best ?
   > (a)"İlişkili değil (*Not related*)"
   > One of the main relations:
   > (b)"rahatlayan (*thing relaxing*)"
   > If **not** any of the above:
   > (c)"Ettiren, yaptıran (*Someone/sth that makes/causes someone/sth to do sth*)"
   > ...

   Though contributors encounter the adjective "rahatlatıcı (*comforting*)" in the sentence, they are asked to decide the relation between the lemma "rahatla (*to relax*)" and "...şeyler (*things*)". Since "...şeyler (*things*)" comfort the person, it is annotated as (c) the causer i.e., ArgA of the relaxing event (see Sec. 2.3.2.1 for causatives). Although it has been considered a challenge, our observation and crowdsourcing results revealed the opposite. Turkish native speakers identified semantic roles easily due to their natural ability to grab the overall meaning of the sentence and abstract away from the syntax.

2. Our focus is to mark the start/end of the arguments as in CoNLL-09 [19] rather than the arguments' full span as in CoNLL-05 [17]. Consider the following example:

   > [Et sevip sevmediğimi]$_{\text{Question}}$ **sor**du. (*She* ***ask****ed me [whether I like meat or not]*$_{Question}$.)

Which defines the relationship of "..sevmediğimi" (*whether I don't like*) with the verb "sor (*ask*)" ?

Annotators are given the last unit of the argument span "...sevmediğimi" (*whether I don't like*) and expected to imagine the full span [Et sevip sevmediğimi]$_{\text{Question}}$ (*[whether I like meat or not]*$_{Question}$.) to choose the best semantic label. Similarly we observed that auto-completion *i.e., imagining the full argument span* was a natural process performed by human mind. Furthermore, marking start/end of semantic roles greatly simplified the aggregation and evaluation processes in contrast to full-span annotations which have been reported to cause problems [61].

3. The number of options is much higher than a typical crowdsourcing task, which may be considered overwhelming for non-experts. In order to address this concern, we have designed different interfaces and launched them on a small data set of 100 questions. In the first design, we have preselected the most likely semantic role label for the argument candidate (the most likely by means of simple statistics based on dependency graph structure and basic morphology). In another design, we first showed the core/numbered arguments (2-3 options on average) with the *Other* option, and only showed the list of adjunct tags if the annotator selected *Other*. For comparison, we have also designed the most basic interface, where we show all options at once without a hint. As opposed to our expectations, the most basic design attracted more taskers and finished the earliest. We explain this phenomena with the following facts according to our observations and feedback from the users:

   (a) Unlike a typical crowdsourcing task, our labels/options are not equally distributed. For instance, total number of core arguments (only A0-A4) is higher than number of all adjunct arguments combined. According to human statistical learning theory, human mind extracts statistical regularities as part of the learning process while executing the task [77, 78]. Therefore, as the annotator answers more questions and gets immediate feedback. He/she builds a probability map of semantic roles given the sentence and the predicate lemma. Since the labels are not equally distributed, in most cases, only a small number of labels have high enough probabilities. Although first few pages look overwhelming and hard to answer for taskers, they adapt immediately

and start building probabilities that help them reduce large amount of options to a viable number of selections as far as we have observed;

(b) Preselection made annotators slower because it increased the average number of decisions to be made (first to decide if the preselected choice is right, then deciding on the correct role). We believe they have built better statistical models without preselection;

(c) A two-level selection was similarly slower, since they needed to click *Other* each time they could not decide on the correct label and wanted to see the list of adjunct tags.

The quality of generated questions can be evaluated by checking the *This question is incorrect* option. The proportion of questions that are checked as incorrect is measured as 0,002. After we have analyzed the comments of crowdworkers, we have come to the conclusion that the errors reported were due to punctuation/spelling errors in the original sentence (original sentence from the treebank) and not related to automatic generation of predicate-argument pairs.

One could argue that, having a specific order for the questions (order by predicate or by sentence) would decrease the context switch of the crowdworkers and help them finish their task earlier. Unfortunately, the crowdsourcing platform we have used, did not have any support for controlling the order of the questions at the time being. [9]. To the best of our knowledge, the platform chooses a question that are not yet answered by 3 different crowdworkers, without considering the previous question shown the tasker.

Another issue for discussion is the average argument counts per predicate. On average, a predicate has 1,80 arguments and 1,20 core arguments in Turkish PropBank, while Chinese PropBank reports 2,92/1,97 arguments and 2,04/1,20 core arguments for verb/nominal predicates respectively; and a verb predicate in the English Propbank averages 3,20 arguments and 2,50 core arguments. The low number of average argument counts per predicate can be explained with two facts. First, having vast amount of nominal predicates (verbal nominals), which are known to have less arguments compared to verbs. Second, being a member of pro-drop (pronoun

[9] Annotators are not constant from the beginning till the end of the task. They can instantly join/exit a task. Therefore imposing a strict order on the task would introduce a synchronization overhead for the platform

dropping) languages which means certain class of pronouns may be elided on the surface. Subject and object pronouns are usually dropped as in the following example:

Gideceğini				söyledim		
Gid	eceğ	i	ni	söyle	di	m
git	FUT	p3s/p3p/p2s	ACC	söyle	PAST	1s
go	FUT	p3s/p3p/p2s	ACC	say	PAST	1s

"**I** said that **you/he/she/it/they** will go"

Here, *söyle-di-m (I said)* is marked with first person singular, which suggests the subject is *I*. Besides, the object pronoun may be 2nd/3rd person singular or 3rd person plural depending on the context. English translation of this sentence would have two predicates *go* and *say*, and three arguments *you/he/she/it/they* for *go*, *I* and the phrase *that you/he/she/it/they will go* for *say*. On the contrary Turkish PropBank would have only one argument *Gideceğini (that you/he/she/it/they will go)* for *söyle (say)* while having same amount predicates, which greatly reduces the average argument count per predicate. Another consequence of missing explicit subject pronouns is the decrease of Arg0 (Agent or Experiencer) proportion in comparison with PropBanks for languages without pronoun dropping property (if corpus properties are similar, e.g., if they consist of newspaper articles or children stories).

3.8 Dataset Statistics

Annotation statistics for the training, development and test splits are given in Table 3.7. Out of 1285 senses, total of 1052 is used in all dataset. Histogram of predicate senses

	Training	*Dev*	*Test*	*All*
#sentence	3947	844	842	5633
#word	39444	8627	8330	56401
#token	44034	9687	9337	63058
#predicate	8151	1834	1757	11742
#distLemma	634	356	333	685
#distSense	960	504	506	1052
#argument	14778	3241	3180	21199

Table 3.7 : Semantic layer statistics on IMST. #distLemma: number of distinct predicate lemmas, #distSense: number of distinct predicate senses

is shown in Fig. 3.15. As can be seen, most of the predicate senses are seen less than 5 times. Argument counts for all splits are given Table 3.8.

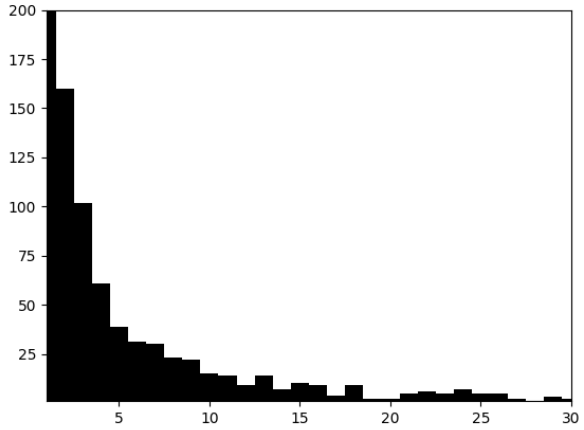


Figure 3.15 : Histogram of predicate senses in training data

	Training	*Dev*	*Test*	*All*
A1	5384	1219	1209	*7812*
A0	2646	587	567	*3800*
AM-TMP	1166	230	218	*1614*
AM-MNR	1049	229	208	*1486*
A2	936	193	201	*1330*
AM-LOC	593	128	138	*859*
AM-LVB	485	109	100	*694*
A4	441	106	74	*621*
AM-GOL	293	66	57	*416*
AM-EXT	259	58	67	*384*
AM-CAU	240	57	62	*359*
AM-ADV	240	39	38	*317*
A3	202	41	46	*289*
A-A	134	21	23	*178*
AM-INS	120	25	22	*167*
AM-PRD	94	16	34	*144*
C-A1	88	18	20	*126*
AM-DIS	87	25	27	*139*
AM-TWO	61	18	13	*92*
AM-COM	60	11	13	*84*
AM-DIR	53	13	13	*78*
AM-NEG	48	8	11	*67*
C-A0	35	7	10	*52*
R-A0	21	4	2	*27*
R-A1	17	3	2	*22*
C-A2	12	6	2	*20*
AM-MOD	4	0	2	*6*
R-A2	2	0	1	*3*
C-A4	2	0	0	*2*
R-A4	1	0	0	*1*
AM-REC	1	0	0	*1*
Total	14778	3241	3180	*21199*

Table 3.8 : Argument counts in Turkish PropBank

3.9 Summary

Semantic role labeling especially for non-English languages suffers from the lack of necessary resources. To fill this gap we have built the first SRL resource for Turkish: **Turkish PropBank**. It is constructed on IMST that contains around 5,6K sentences, is morphologically analysed, POS tagged and manually labeled with shallow and **deep** dependencies.

This chapter has focused on verb sense disambiguation and semantic role annotation of arguments in the treebank with the help of crowds. The annotation framework that harnesses crowdsourcing is described and the role of frame files, plain language in semantic role descriptions and annotated examples in ensuring annotation consistency is discussed. We explained the quality control mechanism based on test questions which enabled us to remove under-performers, continuously train taskers and give real-time feedback. We demonstrated the feasibility of our approach on annotation examples of verbal nominals, nominal verbs and copulas. We provided evaluation results for annotation consistency and interpretation of confusion matrices of semantic roles. We have automatically aligned the semantic layer described in this work with the universal dependency compliant treebank IMST-UD.

4. Statistical Turkish Semantic Role Labeling

This chapter uses the resource we have created through the chapters 2 and 3 to train a statistical classifier for automatic labeling of semantic roles. First we introduce the SRL task, its associated shared tasks and the evaluation technique. We train separate logistic classifiers for PD, AI and AC steps that uses separate set of language specific features. We evaluate our method on Turkish PropBank and perform an error analysis on label predictions with different features.

4.1 Background and Related Work

Most work on SRL have considered SRL as a supervised machine learning problem starting from the first work to tackle SRL as an independent task [79]. The CoNLL shared tasks of 2004, 2005 [17], 2008 [18] and 2009 [19] were devoted to automatic SRL. Traditionally, it is treated as a multi-step classification task. The first step is predicate identification followed by predicate disambiguation. Afterwards, arguments are identified and classified as illustrated in Fig. 4.1. Generally, linear classifiers such as logistic regression and SVM are employed for each step, based on features extracted from the training corpus. Those features heavily rely on syntactic information in the form of a constituency or a dependency parse tree.

Supervised systems require vast amount of training sentences. The scarcity of annotated corpora has motivated the research into semi-supervised learning of semantic representations which utilize both annotated and unannotated data to estimate a model [80, 81]. Knowledge transfer from resource-rich to resource-poor languages via production of annotation or models for resource-poor languages have also been investigated to tackle the paucity of data [82, 83]. In addition to semi-supervised and cross-lingual learning methods, unsupervised techniques that mostly rely on generative modeling and agglomerative clustering have been proposed [84].

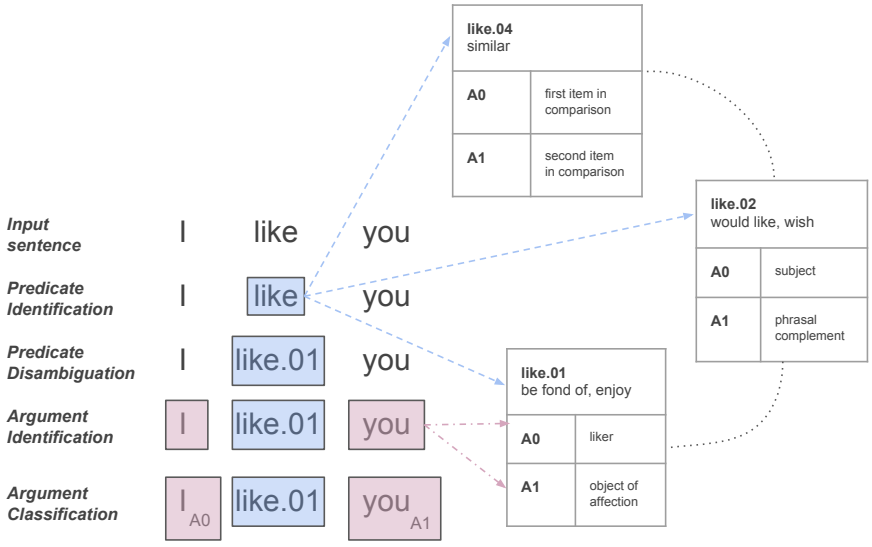


Figure 4.1 : PI, PD, AI and AC steps in SRL pipeline

4.1.1 Data set and Evaluation

There are two types of benchmark data sets. First one aims to assign semantic roles to **syntactic constituents** of predicates and known as CoNLL 2005 ST [16, 17], while the second one assigns labels to **syntactic dependencies** and referred to as CoNLL 2009-ST [18, 19]. Other major differences between CoNLL 2005 and CoNLL 2009 are two fold: CoNLL 2005 does not contain predicate sense information, therefore PD step is omitted; CoNLL 2009 includes nominal predicates [20] *(e.g., inclusion, growth, discussion)*. Since Turkish PropBank has been annotated on a dependency treebank, we use CoNLL 2009 standards. However as discussed in Chapter 3, Turkish PropBank handles nominal predicates via verbs therefore our data does not contain a separate set of nominal predicates. An example sentence taken from Turkish PropBank is shown in Table 4.1. While the lemma, POS, morphological features and dependency information is inherited from dependency treebank, the predicate and argument columns are specific to SRL task. When the token is a predicate, *Pred* column is labeled as *Y* and *Psense* column is filled with predicate sense in that context. There are as many *Arg* columns as number of predicates, indexed according to the predicate order in the sentence. It identifies the semantic role of that token.

ID	Token	Lemma	POS	Feats	Head	Deplabel	Pred	PSense	Arg
1	Bir	bir	Adj	_	2	DETERMINER	_	_	_
2	taksi	taksi	Noun	A3sg|Pnon|Nom	3	OBJECT	_	_	A1
3	bulduk	bul	Verb	Pos|Past|A1pl	0	PREDICATE	Y	bul.01	_
4	.	.	Punc	_	3	PUNCTUATION	_	_	_

Table 4.1 : CoNLL 09 tabular format for SRL

Precision, Recall and F1 are used as the evaluation measure. Dependency-based SRL evaluates both predicate senses and argument labels but only reports the final scores.[1].

$$\text{Precision} = \frac{TP}{TP+FP} \tag{4.1}$$

$$\text{Recall} = \frac{TP}{TP+FN} \tag{4.2}$$

$$\text{F1} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{4.3}$$

True Positive (TP) refers to correctly identified predicate senses and argument labels. Predicate senses are considered correct if sense number matches (lemma information does not matter). Arguments are considered correct if the token id and argument label matches. It should be noted that evaluation script does not take the relation between the frameset and its arguments into account. In other words, if a system assigns the incorrect predicate sense, it still receives points for the arguments correctly assigned. For example, for the prediction *verb.02:* ***ARG0****,* ***ARG1****, ARGM-LOC* and the gold label *verb.01:* ***ARG0****,* ***ARG1****, ARGM-TMP* (for the same tokens), evaluation script calculates a labeled precision score of 2/4 because two out of four semantic labels are incorrect: the predicate sense is detected as *02* instead of *01* and *ARGM-TMP* is incorrectly labeled as *ARGM-LOC*. Scores are calculated for both labeled and unlabeled semantic roles.

4.1.2 Logistic Regression (LR)

Logistic regression is a discriminative classifier that models the $P(Y|X)$ probabilities where X represents discrete features and Y is target labels. LR assumes that this probability can be approximated as a sigmoid function applied to a linear combination of input features, where the sigmoid, a.k.a logit, function is defined as:

$$\sigma(z) = \frac{1}{1+exp(-z)} \tag{4.4}$$

[1]Recent techniques only report argument labeling scores, rather than a joint evaluation

Then the probability for a training point pair (x,y) is assumed to be:

$$z = w_0 + \sum_{i=1}^{n} w_i * x_i \tag{4.5}$$

$$P(Y = 1|X = x) = \sigma(z) \tag{4.6}$$

Under this assumption, the values of W that maximize that probability for all data can be found with gradient descent algorithm. To prevent over-fitting, the parameters are regularized with L_2 regularization, i.e., weights with high values are penalized. Then the general log likelihood is defined as:

$$LL(W) = log \prod_{j} P(y^j|x^j, W) - \lambda ||w||_2^2 \tag{4.7}$$

4.1.3 Reranking

In the early days of SRL, general approach in the field was to train independent classifiers. Therefore the probability $P(Y|X)$ was separately learned for each step: PD, AI and AC. That classifiers are referred to as **local classifier**s, since they assign a label probability of a parse tree node independently from other nodes' labels.

However it is evident that the labels and features of arguments are highly correlated. To address this challenge, Toutanova et.al. [85] proposed modeling a verb's argument structure jointly. They define a log-linear reranking model (given below) that reranks the top N solutions (labels).

$$P(L|t, v) = \frac{e^{<\phi(t,v,L),W>}}{\sum_{j=1}^{N} e^{<\phi(t,v,L_j),W>}} \tag{4.8}$$

where L represents joint assignments to the nodes of the tree t and $\phi(t, v, L)$ denotes a feature map from the tree t, verb v and assignment L. It learns the parameter vector W that maximizes the sum of log-likelihoods of the best assignments. One can also add arbitrary features to the model regarding linguistic constraints. Reranking has been a de facto standard for SRL models based on local classifiers.

4.2 Method

We have adopted the second best system of CoNLL-2009 SRL-only shared task (closed challenge) [86]. The reasons why we have chosen this system are as follows:

- The source code of the system was made available by the authors [2].
- It is easily expandable with new features.
- It does not use the output of multiple syntactic parsers.
- It is fast.

Following the general approach within the field, it sequentially trains independent, local classifiers for PD, AI and AC with L2-regularized logistic regression algorithm and further improves the results with a global reranker [85].

The local classifiers of the adopted system rely on a well-defined set of features which are specifically designed for languages that are part of the original shared task. However none of the languages are from the Altaic family and among participating languages none have an equally complex derivational morphology. In order to address syntactic variations caused by derivational morphology of Turkish, we have implemented features based on case marking, valency information and their combinations with word and POS features that have an obvious importance for Turkish SRL.

Our first method relies on discrete features such as *the count of adjective tokens that are seen as Arg0* that are solely extracted from Turkish PropBank. The complete set of features that have been used by each step is shown in Table 4.3 and definitions of all features are given in Appendix A.3. Although the performance of this approach exceeds our expectations, it still suffers from the data sparsity. Recently, it has been very common to extract additional features from unlabeled data that would contain meaningful statistical information for the task at hand. One of the most popular way of doing this is to use pretrained word embeddings. Although it is still an open research question, whether distributional semantics contain useful information for the task at hand, they have been successfully employed for many different tasks. In order to investigate this issue, we first train word embeddings from a 27 million sentence corpus that we collected, cleaned and compiled. We have used the skip-gram model with hierarchical sampling since it offers better performance for infrequent words. Context window and vector dimension were chosen as 10 and 200 accordingly. To reduce

[2] https://code.google.com/archive/p/mate-tools/

noise, we have only calculated embeddings for words that occur more than 16 times. After training the embeddings, we have introduced continuous features that rely on these embeddings to the SRL system and evaluated their performances on Turkish PropBank.

4.2.1 Features

First set of features are binary and count-based discrete features. We divide these features into five categories, depending on their information level. For instance lexical and positional features are considered low-level features since they are readily accessible without the need of preprocessing. However semantic features are high-level, hard to reach because one requires preprocessing tools to extract those. Ordering from low to high, the features used in this section are roughly as following:

- **Lexical**: Surface form, Lemma
- **Positional**: Distance
- **Morphological**: List of morphological tags, Valency, Case Marker
- **Syntactic**: Parts-of-speech, Dependency label, Dependency path
- **Semantic**: Predicate sense (predicted)

The higher level the feature, the more error is introduced to the system. In other words, while the performance of morphological analyzers may be high enough to employ morphological features, as we climb up the ladders, it is hard to find a robust syntactic and semantic analyzers to extract accurate syntactic and semantic features. For this reason, it is preferable to get competetive results by using low-level features than to get slightly better results employing high-level features.

One of the most useful properties of logistic regression is the possibility of using discrete and continuous features together. [87] propose five continuous features that incorporate distributional word representation trained on raw text with language modeling objective to address the erroneous analysis where syntactic information is not sufficient. They are given in Table 4.2. While the first two features: WordEmbeddingArg and WordEmbeddingPredicate use embeddings directly, other

features aim to model the interaction between predicate and argument and arguments governed by the same predicate.

Feature	Definition
WordEmbeddingArg	$\vec{a}$
WordEmbeddingPredicate	$\vec{p}$
WordEmbeddingCompArg	$\vec{a}+\vec{p}$
WordEmbeddingAvgArg	$\frac{\sum_{i \in n} \vec{w}_i}{n}$
WordEmbeddingPath	$\sum_{w \in deppath(a,p)} \vec{w}$

Table 4.2 : Continuous features based on pretrained word representations proposed by Roth and Lapata [87]

Feature	PI	PD	AI	AC
PredPOS	✓	✓		✓
PredPPOS	✓	✓		
PredLemma+PredDeprel	✓			
PredPPOS+PredPOS	✓	✓		
PredDeprel	✓	✓		
PredFeats	✓			
PredParentPOS	✓	✓	✓	✓
PredParentFeats	✓			
PredParentLemma	✓	✓	✓	✓
PredLemma+PredParentLemma	✓			
DepSubCat	✓	✓		
ChildDepSet	✓	✓		✓
ChildLemmaSet	✓	✓		
ChildPOSSet	✓	✓		
ChildLemmaSet+PredParentLemma		✓		
ChildLemmaSet+PredParentPOS		✓		
ChildLemmaSet+PredPOS		✓		
ChildLemmaSet+ChildDepSet		✓		
ChildLemmaSet+PredLemma		✓		
PredPOS+PredParentLemma		✓		
DepSubCat+PredParentLemma		✓		
ChildPOSSet+ChildDepSet		✓		
ChildCaseMarkerSet		✓		
ChildDepSet+PredLemma		✓		
DeprelPath			✓	✓
ArgPOS			✓	✓
ArgPPOS			✓	✓
ArgPPOS+ArgPOS			✓	
ArgDeprel			✓	✓
POSPath			✓	✓
ArgLemma			✓	✓
Position			✓	✓
PredLemmaSense			✓	✓
ArgDeprel+DeprelPath			✓	

POSPath+RightSiblingPOS			✓	
ArgDeprel+ChildDepSet			✓	
ArgDeprel+ArgPOS			✓	
LeftPOS+RightPOS			✓	
ArgDeprel+PredDeprel			✓	
ArgCaseMark			✓	✓
PredValency			✓	✓
LeftSiblingPOS				✓
LeftPOS				✓
RightPOS				✓
ArgLemma+PredLemmaSense				✓
ArgDeprel+PredLemmaSense				✓
ChildDepSet+Position				✓
ArgDeprel+RightPOS				✓
Position+PredLemmaSense				✓
ArgPOS+ArgLemma				✓
ArgPOS+PredLemma				✓
ArgDeprel+ArgPOS				✓
ArgCaseMark+PredValency				✓
ArgCaseMark+PredLemmaSense				✓
ArgCaseMark+PredLemma				✓
ArgMWE+PredLemmaSense				✓
ArgFirstPosition+ArgLemma				✓

Table 4.3 : Discrete Feature List

4.3 Research Questions

We aim to answer the following research questions by employing the method and the features discussed above.

- (1) In Chapter 2, the importance of morphological features, specially case markers and valency changing morphemes are discussed. Are they really impactful for Turkish SRL?
- (2) Turkish PropBank is built upon a small treebank, that can be considered insufficient to extract meaningful statistics for SRL. Can we estimate how much training data is needed for a system with low generalization error?
- (3) Turkish lacks robust syntactic parsers, hence extracting high-level, robust syntactic features may not be realistic in a real-world scenerio. How many high-level features are actually needed for each step? For instance, is it possible to achieve an acceptable F1 measure by ignoring dependency tree features?

To address the first set of questions, we employ our model with discrete features. As discussed in previous chapter, rich derivational morphology of Turkish results in high OOV rates due to *infinite lexicon* problem. Due to low word coverage rate of IMST, discrete lexical features may not be so helpful. Therefore the second set of questions are:

- (4) Do continuous features offer more than what has already been learned by discrete features?
- (5) Can those features be used as a replacement for high-level features?

4.4 Experiments

We have partitioned the Turkish PropBank based on training, development and test splits of IMST-UD v1.3 distribution [76]. The results have been evaluated with the *eval09* script [3] provided by CoNLL-2009 shared task.

Important statistics about Turkish PropBank, related to PD evaluation are:

1. Similar to other PropBanks 0,66 of lemmas are annotated with their first sense.
2. 42 out of 1757 predicates have not been seen in the training data, so the upper bound is 0,98.

4.4.1 Q1: How important is morphosemantic features?

In Table 4.4, labeled F1 scores for developed systems are given. We have started with a set of language-independent features that are originally designed for English and used by the majority of participating languages and achieved labeled F1 score of 74,91. By incorporating child case marker information for PD and argument case markings for AI and AC, labeled F1 score has been raised to 78,10. It should be noted that case marking information is combined with other features *e.g., ArgCaseMark+PredLemma*. We believe they serve as explicit selectional restrictions to the system.

Valency changes of predicate lemma has improved the labeled F1 score by 0,65. Finally we have substituted all *word (surface)* features with *lemma* features. Although

[3] We have slightly modified eval09 script to account for Universal Dependencies. These changes are only about reading UD files (e.g., different column index for predicate lemma sense) and do not modify the algorithm.

Feature Set	Step(s)	F1	F1-UD
Basic		74,91	73,63
+Case Marking	PD, AI, AC	78,10	77,10
+Valency	AI, AC	78,75	77,43
-Word+Lemma	PD, AI, AC	79,10	77,36

Table 4.4 : Labeled F1 scores for baseline Turkish SRL system. +:addition of feature to previous system; -Word+Lemma:substitution of word features with lemma features. F1: Score of original Propbank; F1-UD: Score of UD compliant PropBank

it did not have a big impact on the labeled F1 score, it has tremendously reduced the model size. We have performed our tests on two different representations: Turkish PropBank based on IMST and IMST-UD. We have used reranking and 5-fold cross validation for both annotation schemes.

The system with the same set of features achieved a labeled F1 score of 77,36 on UD annotation scheme, which is a 1,76pp drop compared to IMST. As reported in study by [76], IGs are in contradiction with the UD idiom therefore derivational morphemes are assigned derivation identifications. For example nouns derived from verbs with INF1 part-of-speech are identified with VERBFORM=GER morphological tag which increases unique morphological features to 67 from 47. Furthermore, the number of unique dependency labels have increased from 16 to 29. We believe the dramatic growth of the feature space play an important role on the performance decrease which can later be addressed by increasing the size of annotated corpus and using more optimized features.

4.4.2 Q2: How much training data is required?

We investigate how number of training instances effect different components of the statistical system. For this purpose, we divide the training set into 10 pieces with equal (x) number of sentences. Afterwards the same model has been trained on multitudes of x and evaluated on the test split. Predicate sense accuracy as a function of x (number of training sentences) is given in Fig. 4.2. The accuracy regularly (almost linearly) increases from 0,74 to 0,83. The slope of the line doesn't seem to increase, which suggests that current size of the training data is not sufficient for predicate disambiguation step.

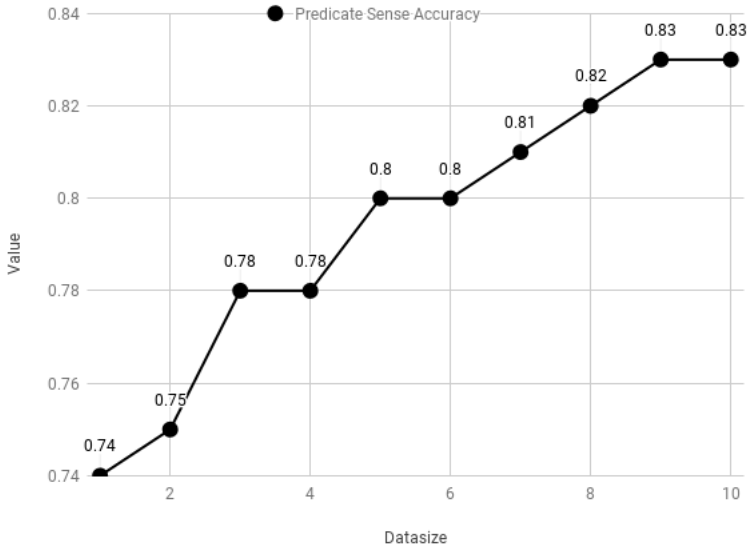


Figure 4.2 : Data size versus PD accuracy

The changes in Precision, Recall and F1 for argument labeling (AI+AC) with respect to the size of training instances is shown in Fig. 4.3. Unlike predicate disambiguation, acceleration is not steady. After training on the first 60% of the data set, performance improvement slows down. Similar trend can be observed in system's

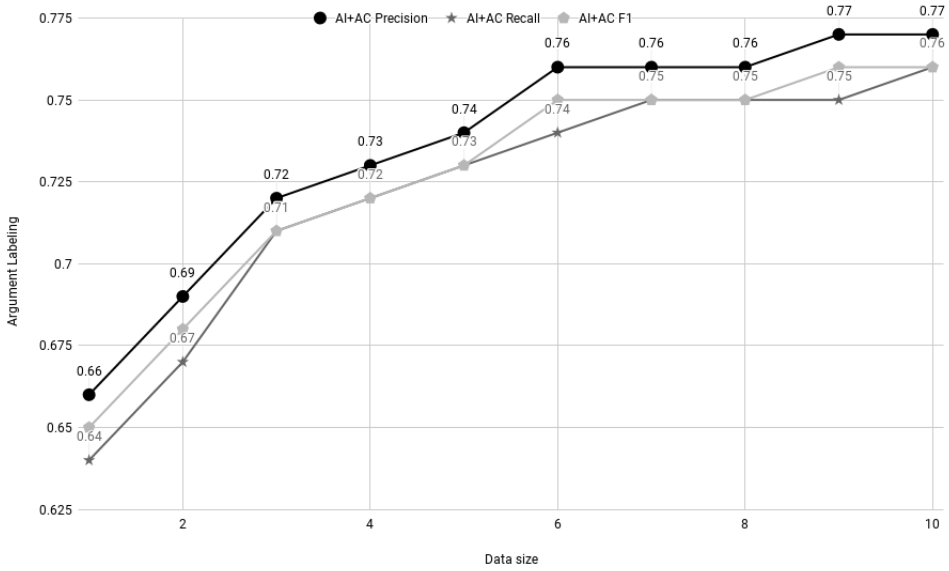


Figure 4.3 : Data size versus argument labeling scores

overall performance shown in Fig. 4.4 and Fig 4.5. First figure illustrates the slowing trend at 6th data batch on overall precision, recall, F1 scores; whereas the second figure

demonstrates the trend on exact semantic match (propositions- all predicate argument pairs- match with gold labeled data).

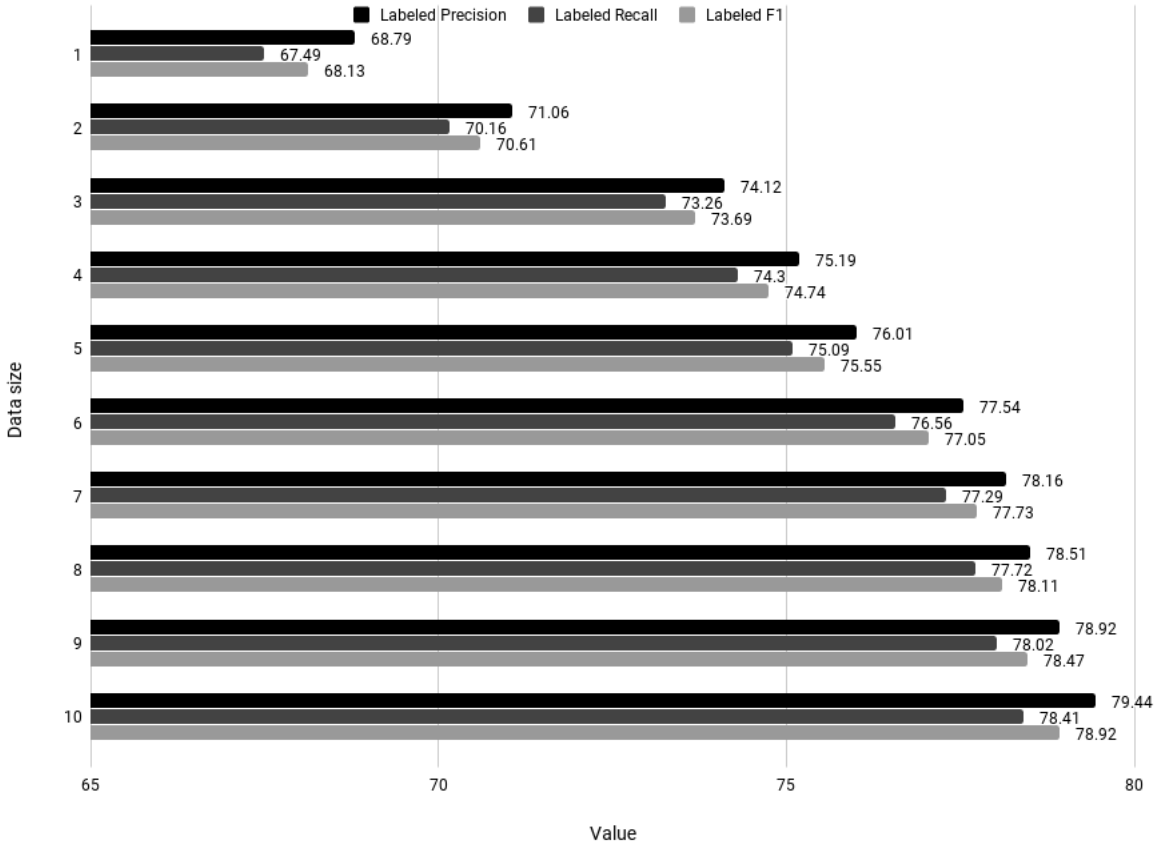


Figure 4.4 : Effect of data size on overall scores

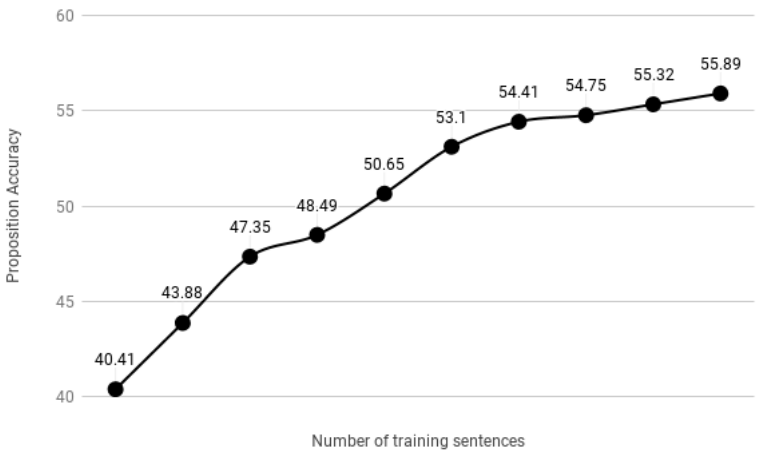


Figure 4.5 : Data size versus fully correct propositions

4.4.3 Q3: Are high-level features actually needed for SRL ?

In order to answer this question, we stacked up features (ordered according to their requirement level) one by one and performed standard evaluation on trained models. It should be noted that, adding morphological level features does not have the same meaning as adding morphological features as in Section 4.4.1. The reason is that a morphological feature may be a composition of morphology and syntactic feature, *e.g., ArgPOS+ArgFeats*. Therefore experiment in this section, **only** employs the features that require maximum of desired level. The results are shown in Table 4.5. This table clearly shows that syntactic level of knowledge is the minimum requirement for a comparative system. Using lexical, positional and morphologic level of information alone could only achieve 7,66 F1 measure of argument labeling; while adding POS features increased that score to 69,16. Furthermore, there are clear performance gains that dependency tree (syntax) and predicate sense lemma (semantic) features can yield to.

		PD	AI+AC Precision	AI+AC Recall	AI+AC F1	Labeled Precision	Labeled Recall	Labeled F1
Lexical	*Lemma*	80,42	36,73	0,57	1,11	79,24	28,99	42,44
	Surface	78,94	46,84	1,16	2,27	77,56	28,84	42,05
+Distance		80,42	37,89	1,13	2,20	78,24	29,35	42,69
+Morph		81,73	36,66	4,28	7,66	73,87	31,84	44,5
+Syntax	*Pos*	82,64	69,42	68,90	69,16	74,15	73,79	73,97
	Dep	83,15	75,94	74,43	75,18	78,54	77,54	78,03
+Semantic		83,00	77,00	76,00	76,50	79,44	78,41	78,92

Table 4.5 : Effects of information level of the features

4.4.4 Q4-Q5: Contribution of Continuous Features

We tackle the question whether distributional semantics (1) can further improve our model with discrete features (2) can be replaced by lexical, syntactic and morphological features. To answer these questions we have designed a simple, focused experiment where we alternate the aforementioned features with continuous features on a case study of argument labeling, which the results are shown in Table 4.6. Similar to previous section, combinations of features are omitted. For instance for the **-Dep+We** (substitute dependency level features with continuous features), we omit

the continuous feature WordEmbeddingPath since it requires the dependency path information.

The results of this experiment not so surprisingly suggest that word embeddings can not be used as a replacement of dependency tree and morphology based features. However the loss is rather small when replaced with lexical features. Surprisingly, when those features are substituted with POS features (and all combinations with POS), argument classification is improved by 0,53 points. This supports the claim that embeddings learn *some* syntax. Finally the original model with discrete features greatly benefited (+0,92 gain in overall F1 scores) from distributional semantics as expected. It can be interpreted as these new features contain complex interactions between information levels that had not been modeled by our previous discrete feature list (Table 4.3).

-/+we	**AL-P**	**AL-R**	**AL-F1**	**OA-P**	**OA-R**	**OA-F1**
We Only	25,96	3,80	6,63	65,00	29,27	40,36
-Dep+We	75,71 -1,29	74,18 -1,82	74,94 -1,56	78,39 -1,05	77,37 -1,04	77,88 -1,04
-Pos+We	77,82 +0,82	76,26 +0,26	77,03 +0,53	79,25 -0,19	78,71 +0,3	79,23 +0,31
-Morph+We	75,80 -1,2	74,28 -1,72	75,03 -1,47	78,45 -0,99	77,44 -0,97	77,94 -0,98
-Lex+We	76,25 0,75	74,72 -1,28	75,48 -1,02	78,74 -0,7	77,72 -0,69	78,23 -0,69
Org	77,00	76,00	76,50	79,44	78,41	78,92
Org+We	78,79 +1,79	77,20 +1,20	78,00 +1,50	80,36 +0,92	79,32 +0,91	79,84 +0,92

Table 4.6 : Effect of the continuous features. AL: Argument Labeling (AI+AC), OA: Overall system (PD+AI+AC), Org: Best system with only discrete features.

4.5 Analysis of Experiments

We carry out analysis on the best model (model with all discrete and continuous features) from previous section. Analysis are given separetaly for PD and AI+AC modules.

4.5.1 PD Analysis

We analyze the effect of derivation types for predicate disambiguation in Table 4.7. Here derivation column accounts for the word type that the predicate derives into. Table shows that adjective derivations are the most challenging cases for the PD module, followed by nouns and adverbs. Further investigation of erroneous PD cases leads to Fig. 4.6 where predicate lemmas with the number of errors overlayed with total number of rolesets are shown. The major error source was found to be the lemma *ol*. This is not surprising considering its syntactic variation, *(e.g., copula, verb, MWE, LVB)*. Also, it is frequently used to as an adjective to build relative clauses, which explains the high error ratio of adjective derivations.

Derivation	False	Correct	Total	**Error**
Adj	58	201	259	**0,22**
Noun	68	285	353	**0,19**
Adverb	16	88	104	**0,15**
Verb	161	880	1041	**0,15**
Total	303	1454	1757	**0,17**

Table 4.7 : PD Errors vs Derivation Types

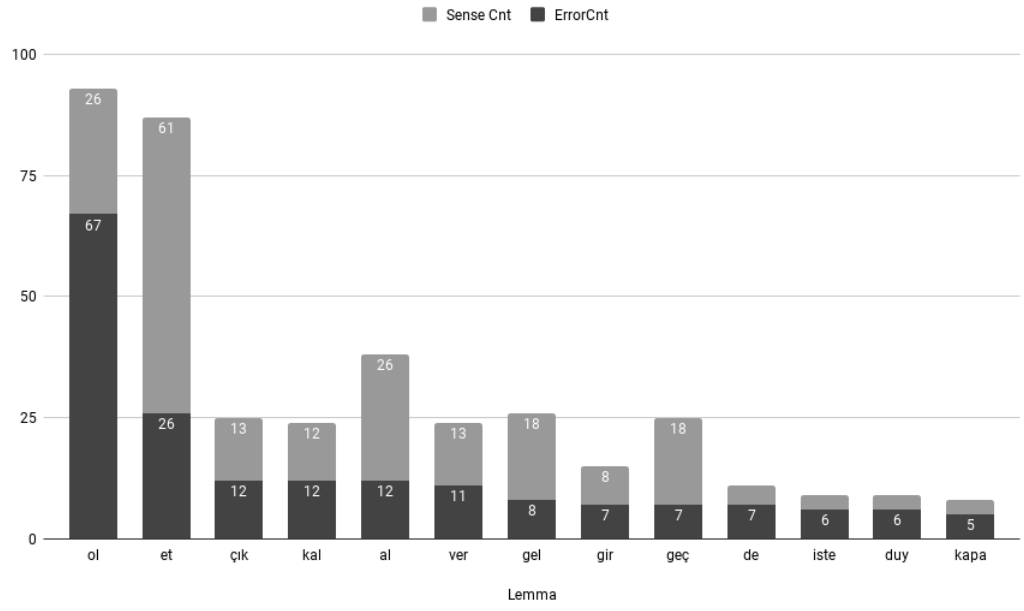


Figure 4.6 : Lemma vs Error

We have investigated the relation between predicate sense confusions and the sentence length shown in Fig. 4.7. The error ratio is calculated as (number of errors for sentences length l)/(total number of predicates for sentences length l). Our analysis shows that there is no clear connection between sentence length and predicate sense disambiguation.

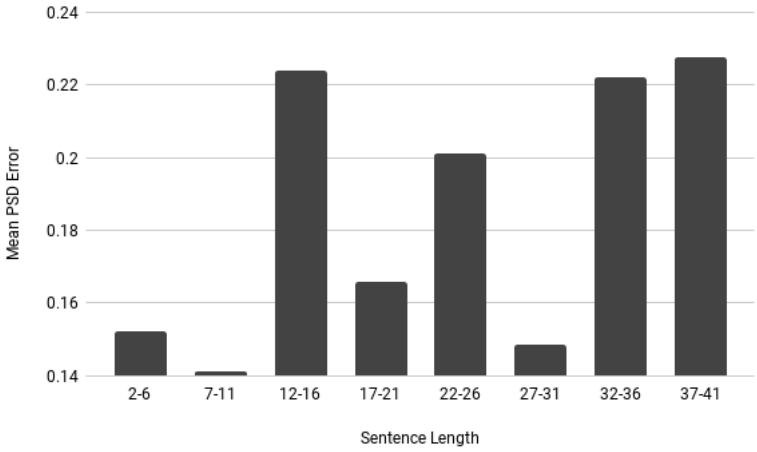


Figure 4.7 : PD error versus sentence length

4.5.2 Argument Labeling Analysis

The performance of each semantic role has been measured separately as given in Table 4.8. It is evident that as the consistency of semantic roles goes down, F1 decreases. For instance, AM-LVB, A1 and A0 consistently refers to light verb argument (usually evident from dependency label), directly affected object (patient, theme) and agent; where AM-ADV (Adverbial) and AM-DIS (Discourse) have vague definitions.

Similar to PD analysis, we investigated if there is a correlation between number of errors and sentence lengths as shown in Fig. 4.8. Although there is a small increasing trend, we believe number of test sentences (only 800 sentences) is not sufficient to draw any conclusions. Finally, we check if there is a connection between labeling errors and their distance to the governing verb. We consider identification and classification errors jointly, i.e., if an argument could not be identified by the system, or the assigned label of the argument is wrong or system identified a token as an argument although it is not. The result is given in Fig. 4.9. Unlike sentence length analysis, there is a clear drop in scores, as the argument moves further away from the predicate.

Label	n	P	R	F1
AM-LVB	100	0,93	0,85	0,89
A1	1209	0,85	0,89	0,87
A0	567	0,81	0,79	0,80
A4	74	0,76	0,81	0,78
AM-NEG	11	1,00	0,64	0,78
AM-TMP	218	0,77	0,78	0,77
AM-LOC	138	0,71	0,81	0,76
AM-PRD	34	0,91	0,62	0,74
AM-MNR	208	0,70	0,75	0,73
A3	46	0,71	0,70	0,70
A2	201	0,72	0,66	0,69
A-A	23	0,71	0,65	0,68
AM-EXT	67	0,69	0,60	0,64
AM-TWO	13	0,58	0,54	0,56
AM-ADV	38	0,62	0,47	0,54
AM-GOL	57	0,59	0,47	0,52
AM-INS	22	0,53	0,45	0,49
AM-CAU	62	0,47	0,40	0,43
AM-COM	13	0,67	0,31	0,42
AM-DIS	27	0,53	0,30	0,38
AM-DIR	12	0,36	0,33	0,35
C-A1	20	0,00	0,00	0,00

Table 4.8 : Argument Labeling performance per category

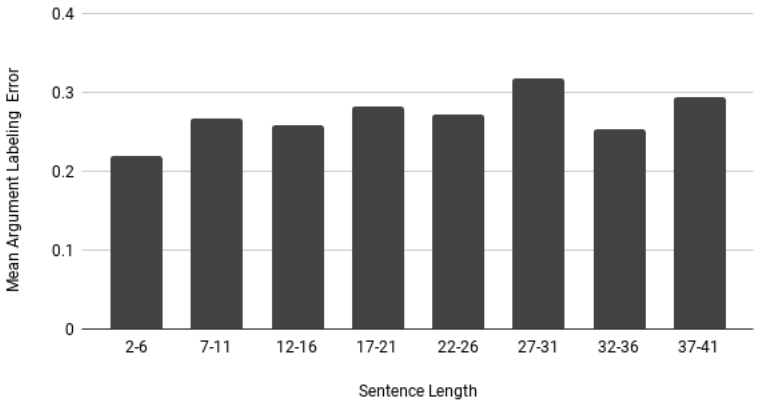


Figure 4.8 : Argument labeling error versus sentence length

4.6 Summary

This chapter has discussed a statistical SRL system based on L2-regularized logistic regression classifiers. First, we have introduced discrete features based on high-level

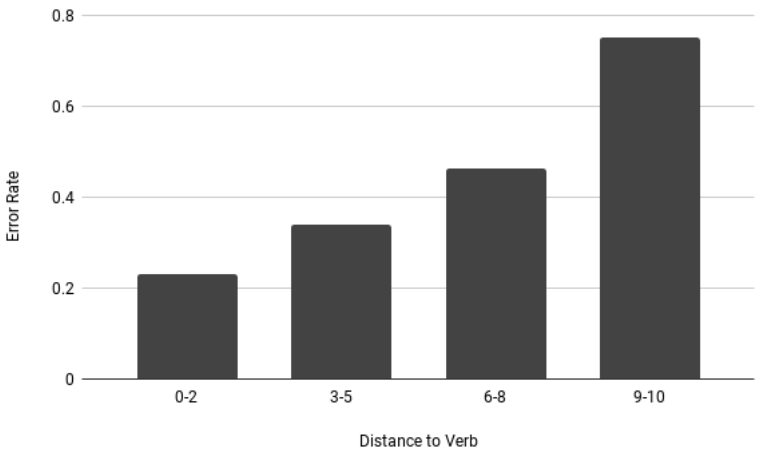


Figure 4.9 : Distance to verb vs argument labeling error

linguistic features such as dependency paths of argument candidates, combination of POStags with predicate's parent's lemma. To address the sparsity of discrete features, we have incorporated continuous features that use pretrained word embeddings. Our experiments with discrete set of features showed that:

1. morphosemantic features are vital for Turkish SRL (specifically case markers);
2. predicate sense disambiguation could benefit from more data more rapidly;
3. an acceptable argument labeling performance can be achieved with well designed features and 60% of the training data;
4. high-level features (at least as high as syntax level) are crucial for the statistical system.

Experiments with continuous features suggested that

1. apart from parts of speech information, they can not be used as a direct replacement for other features such as morphology;
2. performance can further be improved by continuous features which means that they are able to model complex interactions between information levels.

Our best system achieved 79,84 F1 score (without reranking) while the first system with discrete features scored an F1 measure of 78,92. As a comparison, the adopted system had the performance ranging between 76,30 for Japanese and 85,63 for English,

an average of 80,31 labeled F1 score across seven participating languages [19, 86]. Considering differences in annotated corpora sizes (e.g., 13200, 38727, 3948 training sentences accordingly in Catalan, Czech and Turkish), annotation schemes and vocabulary sizes, we can conclude that the 79,84 obtained for Turkish is well within the expected range.

5. Neural Turkish Semantic Role Labeling

The purpose of this chapter is to address the following shortcomings of statistical Turkish SRL discussed in previous chapter:

- relying on external NLP tools for feature extraction,
- need for feature engineering,
- sparsity of extracted features.

Many NLP tasks including SRL has recently seen break-throughs by end-to-end deep learning techniques. Such end-to-end models do not require syntactic preprocessing (or only minimal), *i.e., directly operate on words*, which greatly reduces the required human effort for feature engineering and dependency to lower level NLP tools. Recently introduced neural SRL models report state-of-the-art results with minimal or no linguistic knowledge. However, previous work has been performed on languages with rich resources and less morphological complexity like English and Chinese; and can not be directly applied to a language like Turkish.

In this chapter, we build on previous work on neural SRL and introduce units that are smaller than *words*, referred to as **subwords** or **subunits** to compose *word embedding*s. These units differ by type and information level they require and can be composed via simple or more linguistically motivated techniques. This chapter asks the following questions:

- Is end-to-end neural SRL a plausible technique to achieve or surpass the previous results?
- Is morphology really necessary to identify and classify semantic arguments?
- Do different subword units learn different patterns/complementary information for semantic parsing?

We first give an overview of available neural models followed by motivation for incorporating subword knowledge. Section 5.2 explains subword unit types, word composition methods and the proposed neural SRL architecture. Later we discuss the ways to integrate the knowledge acquired from different subword units. We evaluate various subword unit and composition techniques along with methods to integrate multiple subwords on morphologically rich and poor languages in Section 5.3.

5.1 Background and Related Work

The influence of syntactic features on system performance is discussed in several works and syntactic parser was reported as the major source of classification errors. In order to address this problem, combination of different syntactic parsers and models have been proposed which have reported large improvements on the system. The outputs of different systems are generally combined through an optimization problem. Moreover, most models assume that a syntactic representation of the sentence is available. However for many languages syntactic parsers may have poor quality or are not available at all.

Neural networks have been first introduced to the SRL scene by the frontier work of Collobert et. al [88]. They have presented a unified end-to-end convolutional network architecture to perform NLP tasks such as part-of-speech tagging, chunking, named entity recognition and semantic role labeling. They have reported state-of-the-art results for all tasks except for SRL, emphasizing the complexity of SRL and need for more complex features. Later, a set of work that combine neural networks with traditional SRL features (categorical and binary) have been introduced. Fitzgerald et. al. [89] proposed employing NNs for local role classification after argument candidate detection by another rule based system. The main idea was to embed predicate and argument labels into a shared hidden space $v_{p,r}$, embed argument candidate features in another space v_c and then calculate a score for argument candidate by taking a dot product: $v_{p,r} \cdot v_c$. Later, Roth and Lapata [90], have employed NN to overcome the sparsity of traditional dependency path features both for argument identification and argument labeling. They have used an LSTM network to encode dependency path, i.e., to create a dependency path embedding, and combined it with traditional argument candidate features which were passed onto a nonlinear layer and

later to a softmax layer for label probabilities. Both methods have exploited high level syntactic knowledge from syntactic and dependency parsers; employed NNs only to produce various continuous representations that would be beneficial for SRL task. Furthermore, both methods have only been tested on English dataset.

Following the frontier work by Collobert. et.al [88], it has been shown that careful design and tuning of deep models can achieve state-of-the-art with no or minimal syntactic knowledge [91–95]. Although the architectures vary slightly, they are mostly based on a variation of bi-directional LSTMs. Layers of LSTM are eithern connected in an inter-leaving pattern, as some refer to as *snake LSTM* [91, 95], or regular bi-LSTM layers are more frequently used [92, 93]. Commonly used features for the encoding layer are a type of pretrained word embedding [91–95], distance from the predicate [91, 92], predicate context [91], predicate region mark or flag [91,93,95], randomly initialized POS tags or gold POS tags [92–94] and predicate lemma embedding [93]. All previous works, including the true syntax-agnostic models, state that performance benefit tremendously from gold syntax [91, 95]. From aforementioned models, only works by Diego et.al [93,94] perform dependency-based SRL and all methods focus on languages with rich resources and less morphological complexity like English and Chinese.

Heap's Law [96] defines the relation between the number of distinct units *(e.g., words in the vocabulary)* and units *(e.g., words in a corpus)* as follows:

$$log(|vocabulary|) = w * log(|corpus|) + b \tag{5.1}$$

This empirical law suggests a linear relation between number of units and distinct units in logarithmic space, where w is the rate of the growth and b is the bias. Table 5.1 shows values of w calculated for different languages. As evident from Table 5.1, the

Language	w
Hindi,Urdu	0.55, 0.58
English,Vietnamese	0.63, 0.66
Spanish, Portuguese	0.70
Finnish, Estonian	0.80, 0.82
Hungarian	0.84
Turkish	**0.90**

Table 5.1 : w values of Eq. 5.1 calculated for different languages (Universal Dependency Treebanks (UDT) are used for calculation).

productive morphology of Turkish yields to one of the largest vocabulary growing rates as illustrated in Fig. 5.2. Fast growing vocabulary causes a large number of OOV words in new/unseen datasets. The statistics for vocabulary growth (see Fig. 5.1) and OOV rate (see Fig. 5.2) are calculated from an automatically analyzed corpus of ≈22,2 M sentences, 394M word tokens (surface forms), 3,2M word types and 68K lemma/root types.

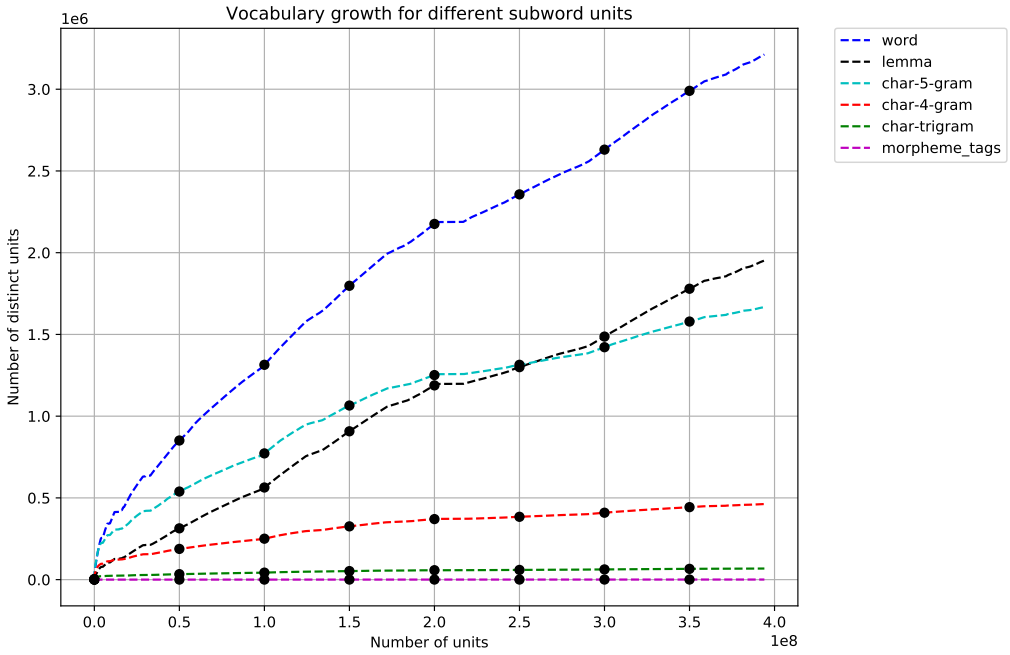


Figure 5.1 : Vocabulary growth with respect to corpus size.

As words are core to any natural language understanding task, OOV problem has attracted considerable attention from the community. In order to reduce OOV rates, using units that are smaller than words, a.k.a. **subword** or **sublexical** units, have been proposed. Some of these subunits and their vocabulary growth on the same dataset is also given in Fig. 5.1. In contrast to word vocabulary, lemma and char-5-gram vocabulary grows at a slower pace; vocabulary of char-4-gram stabilize nearly after $2 * 10^8$ whereas char-3-gram and morpheme tags converge almost immediately.

Wide range of natural language processing tasks such as language modeling, POS tagging and machine translation have benefited from subword units to address rare words [97–102]. Neural language modeling field has shown great interest in composing words from their smaller parts. These models can be investigated in

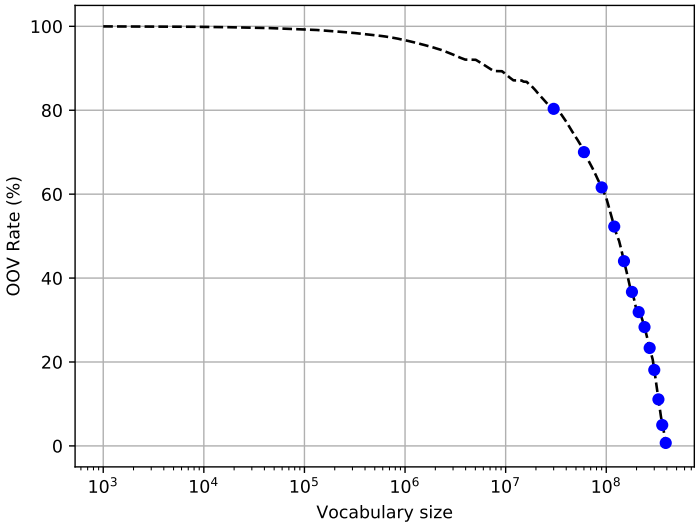


Figure 5.2 : Out-of-vocabulary rate with respect to vocabulary size. OOV rate goes below 40% only when a vocabulary of size 10^8 words is used.

two broad categories: models that require morphological analysis or segmentation and models using only the character information. First set of models integrate morphological knowledge into log bi-linear language models and evaluate the models on semantic similarity tasks [103–105]. Second set use either characters [106, 107], character n-grams [108, 109], bytes [97] or combinations [110, 111]. Various subword units and composition methods have been systematically investigated by Vania et.al [112] on language modeling as a case study. Despite encouraging results of subword units, they have not been investigated on a semantic level to the best of our knowledge. In the light of these findings, we build on previous work on neural SRL by introducing subword compositions instead of pretrained word embeddings and systematically investigate the effects of different subword unit types/compositions and multiple subword integration.

5.2 Method

Previous chapter has concluded that high-level features (at least syntactic) are crucial for the statistical system. However these features require external NLP tools which are either not available or not reliable enough. Recently neural models that can automatically learn continuous representation of units for the task at hand without predefined features are being employed for many NLP tasks. Recurrent Neural

Nets (RNN) are one of such models. They are specially designed for processing sequantial data by passing messages to the next RNN unit over time, i.e., transmitting historical data/context. However, due to difficulties of backpropogation over time (very small/large gradients), it did not yield to satisfactory results in practice. To address the shortcomings of RNN, a specialized network, named Long-Short-Term-Memory (LSTM), have been proposed [113]. In contrast to RNNs, it is capable of learning long-range dependencies, i.e., remembering information for long periods of time.

LSTM networks have been proven to be beneficial for modeling sequential input such as sentences, hence have been widely used for many NLP tasks such as language modeling, named entity recognition and semantic role labeling. Previous neural models on semantic role labeling had a focus on rich-resource languages like English hence were not affected from rare word problem. For that reason they have used "words/tokens" as the smallest meaning bearing unit for SRL. In contrast to previous methods, we investigate different smaller units explained in Section 5.2.1, composed in various ways defined in Section 5.2.2.

Labeling of semantic roles for the sentence *Uzanıp mektubu aldı (He/she reached out and grabbed the letter.)* and the predicate *al (to grab)* by the proposed model is shown in Fig. 5.3. ρ represents the partitioning function that splits the word into subwords, where ϕ denotes the composition function that generates the word embedding, $\vec{w}$, from subword units. Calculation of $\vec{w}$ given in Equation 5.2 is described in details in following sections. There may be more than one predicate in the sentence so it is crucial to inform the network of which arguments we aim to label. In order to mark the predicate of interest, we concatenate a predicate identifier *(1 if the token is predicate, 0 otherwise)* to the word embedding vector. It serves as input to LSTM layer and denoted as $\vec{x}$. Since the input is sequential t is used to refer to the input at time t.

$$\vec{w} = \phi(\rho(w)) \tag{5.2}$$

$$\vec{x_t} = \vec{w} \oplus p_t \tag{5.3}$$

Internal structure of an LSTM unit is given in Fig. 5.4. Arrowed lines correspond to vector transfer and intersecting lines refer to concatenation as in the joint between h_{t-1} and x_t.

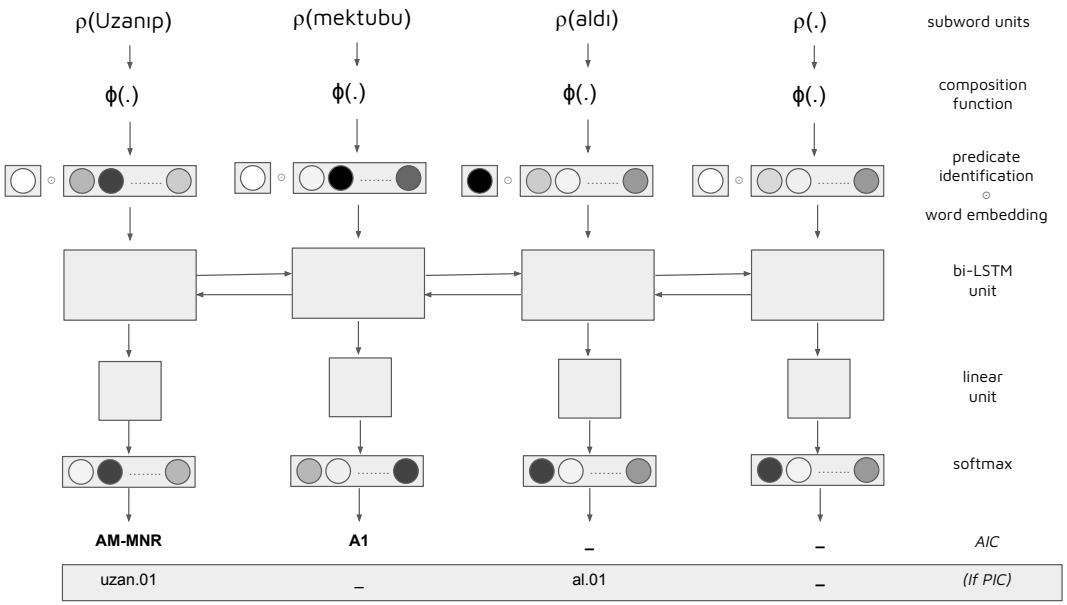


Figure 5.3 : Bidirectional LSTM model for SRL

Each LSTM unit receives and gives two different feedbacks, namely as *cell* and *hidden* states. Cell state can be referred to as the memory of LSTM units that is transmitted over time. Internal mechanism of LSTM is designed to decide which information to keep, add or remove to/from the cell state. LSTM achieves this via four essential neural network layers namely as forget, input, candidate and output. These layers are sometimes referred to as gates since they function as gates that let the information go through. As a matter of fact, they are simple feed forward networks with a σ function as the last layer that maps values between 0 and 1, which are later used to mask inputs. Forget gate layer decides which information to remove from the previous cell state by considering the values of the input $\vec{x_t}$ and the previous hidden state $\vec{h_{t-1}}$ as given in Equation 5.4.

$$\vec{forget_t} = \sigma(W_{if} \cdot \vec{x_t} + \vec{b_{if}} + W_{hf} \cdot \vec{h_{t-1}} + \vec{b_{hf}}) \tag{5.4}$$

In order to decide the new information that should be added to the memory of an LSTM, two different layers are used. First layer is called the Input gate and outputs values between 0 and 1 similar to forget gate by looking at the values of $\vec{x_t}$ and $\vec{h_{t-1}}$. It simply decides which cells of the memory to update. Other layer creates a candidate

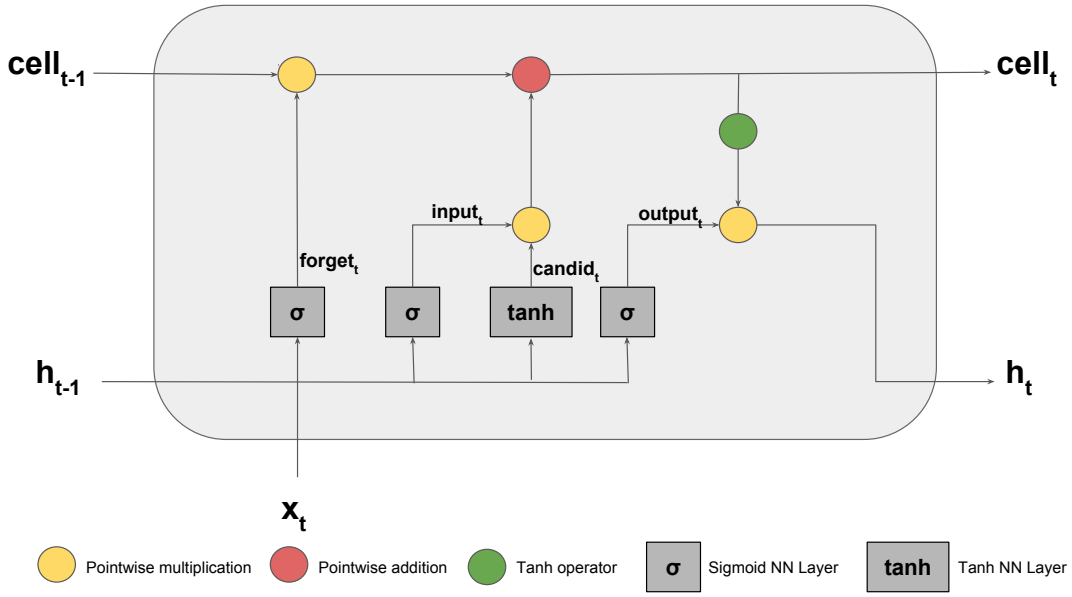


Figure 5.4 : Internal structure of an LSTM unit

memory vector that is mapped between -1 and 1 value range by a tanh function.

$$\vec{input}_t = \sigma(W_{ii} \cdot \vec{x}_t + \vec{b}_{ii} + W_{hi} \cdot \vec{h_{t-1}} + \vec{b}_{hi}) \tag{5.5}$$

$$\vec{candid}_t = tanh(W_{ig} \cdot \vec{x}_t + \vec{b}_{ig} + W_{hg} \cdot \vec{h_{t-1}} + \vec{b}_{hg}) \tag{5.6}$$

Later, the memory of the LSTM is updated with the following equation. It simply does the following: (1) mask the previous cell state with the output of the forget gate, (2) mask/scale the new information with the output of the input gate and (3) add them.

$$\vec{cell}_t = \vec{forget}_t * \vec{cell}_{t-1} + \vec{input}_t * \vec{candid}_t \tag{5.7}$$

Cell states are not directly used as LSTM outputs. First a mask is calculated via an *output* gate, similar to *forget* and *input* gates. It is then multiplied by the cell state to filter out the undesired memory parts. This final value is referred to as the *hidden* state of LSTM unit and denoted as $\vec{h}_t$.

$$\vec{output}_t = \sigma(W_{io} \cdot \vec{x}_t + \vec{b}_{io} + W_{ho} \cdot \vec{h_{t-1}} + \vec{b}_{ho}) \tag{5.8}$$

$$\vec{h}_t = \vec{output}_t * tanh(\vec{cell}_t) \tag{5.9}$$

The complete process described above is usually referred to as one-pass over the sequence. It can be performed in two directions: forward and backward. Forward pass processes the sentence in natural order, while backward pass does so in reverse

direction. This type of unit is called a bidirectional LSTM (bi-LSTM) unit. Recently, it has been shown that processing a sentence in both directions outperforms previous methods in most NLP tasks. Therefore we employ bi-LSTMs instead of standard ones. The outputs from both directions, denoted as $\vec{h_f}$ and $\vec{h_b}$, are generally concatenated for the next layer.

We finally pass the hidden states of the LSTM to a linear layer in order to map down (usually down, can be up too) the values into the semantic role space. Finally, each label's probability is calculated via a softmax function and the label with the highest probability is assigned to the input token.

$$\vec{l_t} = W_{lin} \cdot [\vec{h_f} \oplus \vec{h_b}] + \vec{b_{lin}} \tag{5.10}$$

$$P(label|w_0, w_1, .., w_t, .., w_n) = softmax(\vec{l_t}) \tag{5.11}$$

The same architecture is used for both PIC (predicate identification and classification) and AIC (argument identification and classification) with minor differences. In PIC settings, each sentence is fed to the system once while in AIC task, each sentence is fed multiple times (one for each predicate in the sentence).

The official SRL shared task readily provides predicate identification information. For that reason, we design two PIC systems: (1) pd only (2) joint. First system solely focuses on predicate sense disambiguation and provided with binary predicate markers, while the second one is more similar to AIC, i.e, jointly decides if a token is predicate and if so the sense number.

5.2.1 Subword Units

The idea of sublexical modeling is to represent words/forms as a combination of smaller units so as to decrease vocabulary size and be able to represent unseen words with its subparts. Most commonly used subword units in literature are characters, word parts/segments and morphological analysis acquired by an oracle model. For demonstration of different units, representation of word *İncelenirs (if examined)* is given in Table. 5.2.

char function simply splits token into its characters. Similar to n-gram language models, char-n-gram functions slide a character window of width *n* over the token. Start and end of the token is indicated by special characters. Unlike character level

ρ	Representation
char	<-İ-n-c-e-l-e-n-i-r-s-e->
char-3-gram	<İn-İnc-nce-cel-ele-len-eni-nir-irs-rse-se>
char-5-gram	<İnce-İncel-ncele-celen-elene-lenir-enirs-nirse-irse>
morfessor	İ-nce-lenirse
bpe	İ-n-ce-len-irse
oracle	incele+Verb ∧DB+Verb+Pass+Pos+Aor ∧DB+Adj+Zero ∧DB+Verb+Zero+Cond+A3sg

Table 5.2 : ρ functions and outputs

functions, morpheme functions generally rely on prior knowledge of language to segment words into subwords. General idea behind morfessor is to find a set of morphemes that describe the provided training corpus efficiently and accurately [114]. It is achieved by finding the parameters of the segmentation model that maximizes the probability of training data. **Byte Pair Encoding (BPE)** is originally a data compression algorithm that iteratively replaces the most frequent pair of bytes with an unused byte. After its successful adaptation to language by Sennrich.et.al [100] for addressing rare words in MT, it has become one of the standard subword units. We use both methods with their default parameters. Final representation, oracle, is basically the output of an available morphological analyzer. In case of Turkish, we use the same IG based morphological representation as in IMST [70].

5.2.2 Composition Methods

Let us denote the output of $\rho(w)$ as $s_0, s_1, .., s_n$, where each s_i is a subword of word w given the function ρ. These subunits can be composed in various ways. The most naive way is to add these units together:

$$\textbf{Addition} : \vec{w} = \sum_{i=0}^{n} (\rho(w)) \tag{5.12}$$

Recently, a composition method based on bi-LSTMs has been proposed [106]. It treats words as a sequence of subword units and performs a forward and a backward pass on the sequence as given by the equations from 5.4 to 5.9. Finally it learns a set of weights for each direction and outputs a weighted vector which will be used as the word embedding. Denoting the final hidden state of the forward and the backward pass as h_f and h_b accordingly, $\vec{w}$ is calculated as:

$$\textbf{bi-LSTM} : \vec{w} = W_f \cdot h_f + W_b \cdot h_b + b \tag{5.13}$$

Adding subwords, ignores any kind of order among subword units while bi-LSTM composition assumes subwords follow a strict order. However, for languages with rich derivational morphology such as Turkish, the importance of the morphological tag order varies with the type of morphology. Therefore we propose a novel composition method that is more convenient for inflectional group (IG) formalism. Consider the morphological analysis of the word *Konuşulanları (the things that are being spoken)* given in Fig. 5.5. After splitting the word from derivational boundaries, we end up with three different parts-of-speech types: *Verb*, followed by *Adjective* and finally *Noun*. If we change the order of rectangles *(e.g. Verb->Noun->Adj)*, we would end up with an irrelevant word *(e.g. konuş-ma-cı (speaker))*[1]. Therefore there is a strict order in derivational morphology. The tags other than parts of speech inside each rectangle are to indicate inflections. Unlike derivational morphemes, the order of inflections are fixed hence does not provide an extra information about the word's meaning. For instance if there is an accusative marker and a plurality morpheme, case marker will always follow the plurality morpheme. Therefore there is only one way to say that the word is a definite plural noun. This suggests that adding inflectional morphemes should provide enough information about the meaning. Taking these facts into account,

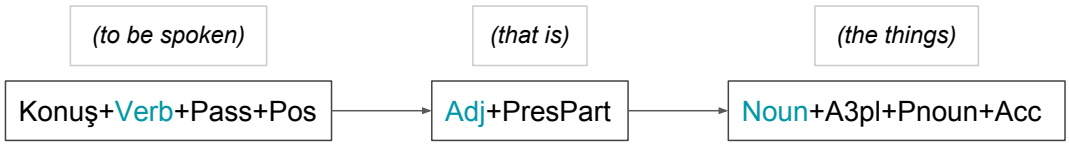


Figure 5.5 : Order in inflectional and derivational morphology

we compose inflectional morpheme tags with addition and derivations via bi-LSTM. Considering the word has n derivational boundaries and each inflectional group has $k_0, k_1, .., k_n$ number of inflectional tags is calculated as following:

$$\textbf{add-bi-LSTM} : \vec{derivation}_i = \sum_{j=0}^{k_i} inflection_j \tag{5.14}$$

$$\vec{h_f}, \vec{h_b} = bi-LSTM(derivation_0, derivation_1, .., derivation_n) \tag{5.15}$$

$$\vec{w} = W_f \cdot h_f + W_b \cdot h_b + b \tag{5.16}$$

[1] In this example, it is not possible to generate a word with exact same tags. Therefore *konuşmacı* is given just as an example for Verb->Noun->Adjective derivation with the lemma *konuş*

In the rest of this thesis, the inplut splitted from DB will be referred to as **oracle-DB** and the proposed composition will be denoted as **add-bi-LSTM**.

5.2.3 Multiple Subword Units

Hypothetically, each input representation dictates a certain set of assumptions which would lead to different error types and different models converging to a different solution. A more robust model can be achieved by combining these base models in a smart way, in exchange for extra time and space complexity. However there is no guarantee that the combined model would be more robust than base models. Consider a set of base models that are not diverse, i.e., making similar errors with similar inputs. In such a case, combined model would not be able to overcome the learners' biases, hence would not yield to better resuls.

We hypothesize that in case of complementary knowledge among these base learners that are trained on different input representations, overall performance of the system would increase when they are combined. In order to test this hypothesis, we integrate different units at the word encoding stage and later at the ensembling stage.

5.2.3.1 Apriori Integration

We alter the input encoding layer as shown in Fig.5.6. Given the word embeddings $\vec{w}_0,..,\vec{w}_i,..,\vec{w}_n$ composed from a single subunit as discussed in Section 5.2, the final word embedding, $\vec{w}_f$, is calculated as $\gamma(\vec{w}_0,..,\vec{w}_i,..,\vec{w}_n)$, where γ is one of the following: SUM, WEIGHTED SUM, MAX and CONCATENTATION. concatenating all vectors may not be theoretically appropriate since it may refer to sampling from *one* multivariate statistical distribution [115]. Furthermore, dimension decreasing composition MAX may oversimplify the input and cause important information loss. In general, any type of apriori integration complicates the learning process, *(e.g., model needs to learn the best representation for char and char3 grams such that their sum would predict semantics better)*, hence may require more data.

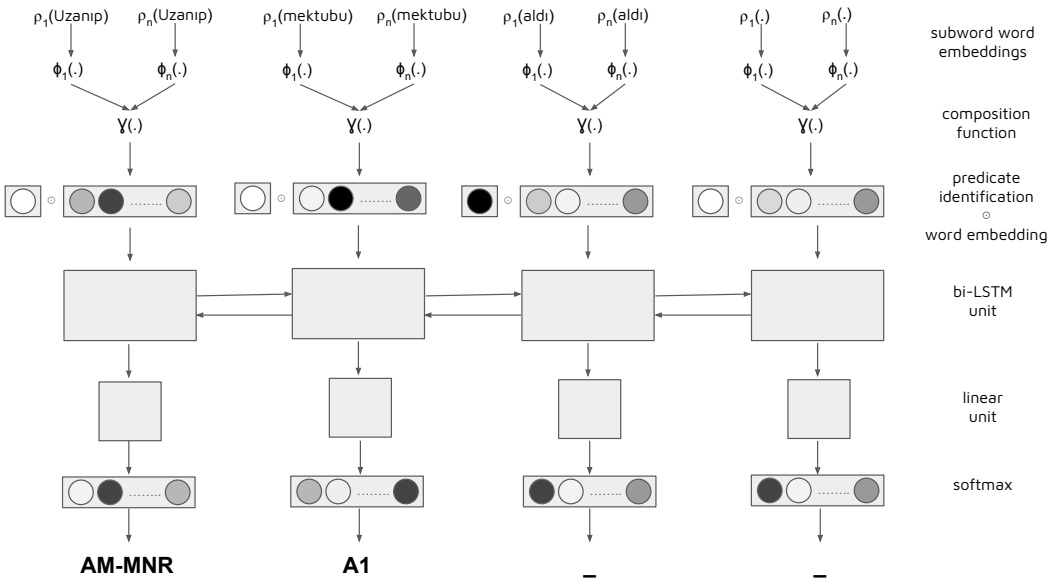


Figure 5.6 : Apriori integration of multiple subword units

5.2.3.2 Post Integration

Suppose that a prediction p_i is generated for each token by a base model m_i, where $i = n$, then the final prediction is calculated from these predictions by:

$$p_{final} = f(p_0, p_1, .., p_n|\phi) \tag{5.17}$$

where f is the composition function and ϕ denotes the parameters. f can be chosen from composition techniques such as sum, weighted sum and maximum introduced in Section 5.2.2. One can also use median, minimum or product of prediction list to decide on the final label. This approach is known as the global combination where all learners generate an output and f uses each output to decide on the final class. The simplest global approach is **averaging**, where f is simply the mean function and p_is are the log probabilities. It has been previously shown that, averaging over models with large variance yields to better fit than individual models. Therefore improvement in the scores would support our hypothesis of diverse learners.

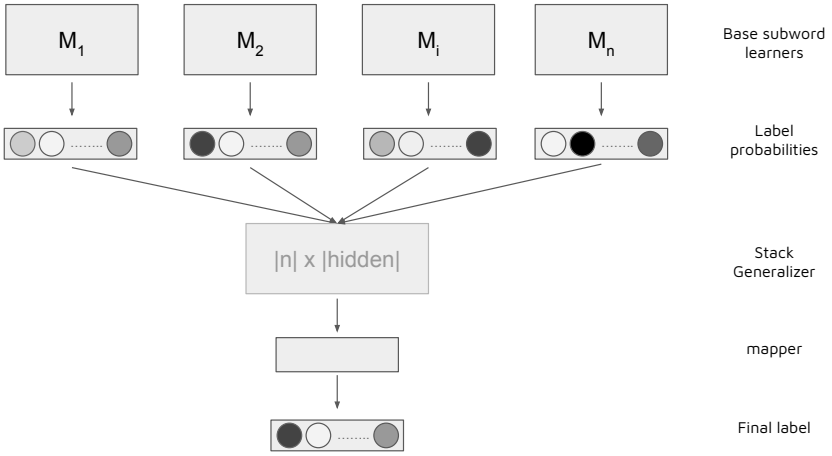


Figure 5.7 : Stack Generalization

A function as simple as mean, may already result with better classification scores. However it combines model outputs linearly, therefore ignores the nonlinear relation between base models. In order to exploit nonlinear connections, we learn the parameters ϕ of f via a simple linear layer followed by sigmoid activation. In other words, we train a new model that learns how to best combine the predictions from subword models. This ensembling technique is generally referred to as stacking or stacked generalization and is illustrated in Fig. 5.7. Given the output combination from the base models, combiner learns the corresponding correct labels. One drawback of stack generalization is the neccesity of a decent sized development set since stacking needs to estimate and correct the biases of the base models on unseen data.

5.3 Experiments

We first measure the performance of simple subword units with various composition techniques on argument labeling benchmark described in Fig. 5.4. Afterwards the effect of apriori and post integration is tested. In order to draw more general conclusions, we perform similar experiments on German, Spanish, Catalan, Czech and Finnish.

5.3.1 Single Unit: Turkish

We have used the same hyperparameters for all combinations given in Table A.9. Due to small training data size, we have combined the training and development splits and used 10-fold cross validation during training. We have used dropout, gradient clipping and early stopping to prevent overfitting. SGD has been used as the optimizer. The initial learning rate is set to 1 and reduced by half if scores on development set do not improve after 3 (patience parameter) epochs. Weight parameters are initialized uniformly between -0,1 and +0,1. We have not performed addition on chars since the char vocabulary is not large enough. Add-bi-LSTM requires data annotated with IG formalism, therefore could only be used for the input splitted from derivational boundaries.

First, we give the results for predicate sense disambiguation with pd only settings. Baseline method assigns the first sense to each predicate lemma. Word model employs pretrained word embeddings instead of subunits and tunes them during training. The results are given in Table 5.3. The results when predicate identification information is

baseline	65,73	
word	69,22	
char	81,24	
oracle-DB	82,05	
	Addition	**Bi-LSTM**
char3	80,43	80,38
char5	77,86	80,43
bpe	76,09	78,15
morf	75,34	72,83
oracle	82,04	**82,78**

Table 5.3 : PD accuracy results for for different subword units composed with addition, bi-LSTM and add-bi-LSTM.

not supplied, is given in Table 5.4.

The labeled and unlabeled argument labeling scores are given in Table 5.5 and Table 5.6 respectively, In general, bi-LSTM composition outperforms addition for any type of subword unit, Morpheme segmentation methods yield to the lowest labeled F1 scores however have a competetive performance on argument identification as can be inferred from unlabeled scores.

	P		R		F1	
word	55,58		39,30		46,05	
char	77,07		68,65		72,62	
oracle-DB	80,44		74,60		77,41	
	Addition			**Bi-LSTM**		
	P	**R**	**F1**	**P**	**R**	**F1**
char3	76,29	64,99	70,19	78,12	72,71	75,32
char5	73,90	50,86	60,25	72,29	69,11	70,66
bpe	74,63	49,49	59,51	73,81	63,04	68,00
morf	72,02	35,93	47,94	69,06	42,91	52,93
oracle	78,70	72,94	75,71	81,33	74,77	**77,91**

Table 5.4 : Joint PI+PD results for different subword units composed with addition, bi-LSTM and add-bi-LSTM.

	P		R		F1	
char	0,61		0,56		0,58	
oracle-DB	0,64		0,58		**0,61**	
	Addition			**Bi-LSTM**		
	P	**R**	**F1**	**P**	**R**	**F1**
char3	0,58	0,27	0,37	0,62	0,54	0,58
char5	0,55	0,29	0,38	0,56	0,46	0,51
bpe	0,55	0,36	0,44	0,55	0,45	0,50
morf	0,49	0,30	0,37	0,55	0,43	0,48
oracle	0,66	0,51	**0,58**	0,64	0,57	**0,60**

Table 5.5 : Labeled argument labeling scores for different subunits and composition functions, Best F1 for each composition is shown in bold.

	P		R		F1	
char	0,80		0,74		0,77	
oracle-DB	0,87		0,79		**0,83**	
	Addition			**Bi-LSTM**		
	P	**R**	**F1**	**P**	**R**	**F1**
char3	0,75	0,34	0,47	0,80	0,66	0,72
char5	0,72	0,38	0,50	0,79	0,66	0,72
bpe	0,81	0,52	0,64	0,81	0,67	0,73
morf	0,70	0,43	0,53	0,80	0,64	0,71
oracle	0,87	0,67	**0,76**	0,84	0,76	0,80

Table 5.6 : Unlabeled argument labeling scores for different subunits and composition functions. Best F1 for each composition is shown in bold.

As expected, SRL benefits from morphological analysis. There is a 3 point difference between the best character model and the best oracle model. Taking the size of the dataset into consideration, 3 point can be considered as small. Furthermore add-bi-LSTM performs slightly better than composing morphological

tags via bi-LSTM. The difference between the scores, originates from derivational boundaries (DB). We believe as the number of DBs increase the gap between scores would be underlined as well. As shown in Table 5.7, average number of derivations per sentence is only 1,18.

#DB	#Sent	#DB	#Sent
0	2209	6	65
1	1251	7	42
2	564	8	20
3	322	9	10
4	203	10	4
5	99	11	1
		13	1
Average		**1.18**	

Table 5.7 : Number of sentences per derivational boundary count

5.3.2 Single Unit: German, Spanish, Czech, Catalan, Finnish

We repeat the above experiment on different languages for which an annotated SRL data is available. The results are given in Table 5.8 for German, Spanish, Czech, Catalan and Finnish. Similar to Turkish experiments, best F1 scores, given in bold, are always achieved by composing morphological units via bi-LSTM. For some languages, character bi-LSTM models perform competitively with the oracle models, while for morphologically rich languages (here Czech and Finnish), oracle information seems to be crucial for semantics.

		Addition			Bi-LSTM		
		P	**R**	**F1**	**P**	**R**	**F1**
German	**char**	-	-	-	0,61	0,61	0,61
	char3	0,63	0,60	0,61	0,66	0,62	*0,64*
	char5	0,49	0,42	0,45	0,58	0,53	0,56
	oracle	0,65	0,63	0,64	0,68	0,64	**0,66**
Spanish	**char**	-	-	-	0,69	0,65	0,67
	char3	0,67	0,67	0,67	0,71	0,66	*0,68*
	char5	0,55	0,52	0,54	0,63	0,49	0,55
	oracle	0,67	0,67	0,67	0,68	0,69	**0,69**
Czech	**char**	-	-	-	0,79	0,73	0,76
	char3	0,71	0,68	0,69	0,80	0,74	*0,77*
	char5	0,68	0,62	0,65	0,76	0,61	0,68
	oracle	0,78	0,74	0,76	0,84	0,78	**0,81**
Catalan	**char**	-	-	-	0,71	0,67	0,69
	char3	0,70	0,70	*0,70*	0,72	0,66	0,69
	char5	0,59	0,57	0,58	0,69	0,52	0,59
	oracle	0,69	0,70	**0,70**	0,71	0,69	**0,70**
Finnish	**char**	-	-	-	0,72	0,61	0,66
	char3	0,66	0,65	0,66	0,70	0,64	*0,67*
	char5	0,59	0,55	0,57	0,64	0,54	0,59
	oracle	0,68	0,66	0,67	0,74	0,71	**0,72**

Table 5.8 : Labeled argument labeling scores on other languages for different subunits and composition functions, Best F1 for each composition and language is shown in bold. Best F1 that do not require oracle information is shown in italics.

5.3.3 Multiple Units: Turkish

The results of our single unit experiments suggest that the morpheme representations are not accurate enough, hence may hurt the performance when combined with other units. Therefore we restrict our experiments with char, char3-gram, char5-gram, oracle and oracle-DB models. Previous works on language modeling suggest that pretrained word embeddings for high frequent words would be better at capturing semantics, whereas character embeddings would capture syntax and morphology. In order to test this, we incorporate pretrained word embeddings (*word*) (see Chapter 4 for settings) into our experiment as another unit.

First experiment aims to test whether apriori integration (see Section 5.2.3.1) of units help the argument labeling scores. Due to computational complexity of apriori integration, we choose *char3-gram* as a representative for character based models and oracle as a representative of morphological models. The results of apriori integration

with various composition functions is given in Table 5.9. As can be seen, pretrained word embeddings alone yield to an F1 score of $0,44$. We have only used the embeddings for high frequent words and embeddings are tuned for the task at hand during training. The reason behind such a low score is the rare word problem as discussed in the beginning of this chapter.

The results suggest that character and oracle representations help each other when combined via maximum or concatenation operation at an early stage. Another result is that the maximum model is somewhat robust to low quality input. Although word model is not as accurate as desired, whn integrated to char3+oracle combination, it did not cause a big drop on F1 scores. In addition, we observe that sum and weighted sum either hurt or does not improve the results. Next, we experiment with the simplest

	Sum			**Weighted Sum**			**Max**			**Concatenation**		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
char3	0,62	0,54	0,58	0,62	0,54	0,58	0,62	0,54	0,58	0,62	0,54	0,58
oracle	0,64	0,57	0,60	0,64	0,57	0,60	0,64	0,57	0,60	0,64	0,57	0,60
word	0,50	0,39	0,44	0,50	0,39	0,44	0,50	0,39	0,44	0,50	0,39	0,44
char3+oracle	0,63	0,57	0,60	0,64	0,57	0,60	0,65	0,58	0,61	0,64	0,59	**0,62**
char3+oracle+word	0,62	0,57	0,60	0,59	0,53	0,56	0,62	0,56	0,59	0,63	0,57	0,60

Table 5.9 : Apriori integration results of *char3*, *oracle* and *word*. First three rows are provided as a reference.

ensembling technique: averaging the log probabilities of different base models, a.k.a product-of-the-experts. The results are given in Table 5.10. Averaging as a post

Combination	**P**	**R**	**F1**
char/char3	0,61	0,56	0,58
char+char3	0,66	0,59	0,63
char+char3+char5	0,69	0,58	0,63
oracle	0,64	0,57	0,60
oracle-DB	0,64	0,58	0,61
oracle+oracle-DB	0,70	0,64	**0,67**
char3+oracle	0,69	0,61	0,65
char3+oracle+word	0,70	0,53	0,61
all	0,73	0,62	**0,67**

Table 5.10 : Ensemble of base models via averaging. Models in italics are provided as references.

integration method yields to a score of **0,67** when two oracle models are combined. Although both models use a similar representation, they use a different composition technique which introduces variance among two methods. On the other hand, unlike

apriori integration, it is not as robust to low quality input. As can be seen from scores char3+oracle+pwe combination, scores drop dramatically after pwe model is introduced.

Next, we evaluate the idea of stack generalization (SG) and compare it with the results of products-of-experts ensembling. To train the SG model, we have used one linear layer with 64 hidden units. Weights are orthogonally initialized and optimized via adam algorithm with a learning rate of 0,02 for 25 epochs. The results are presented in Table 5.11. In contrast to our expectations, the results achieved by SG and averaging are extremely close. Both techniques suggest that results are greatly improved when (1) oracle models with bi-LSTM and add-bi-LSTM are combined; (2) char and char3 models are combined and (3) char3 and oracle are integrated.

Combination	**P**	**R**	**F1**
char/char3	0,61	0,56	0,58
char+char3	0,67	0,59	0,63
char+char3+char5	0,68	0,59	0,63
oracle	0,64	0,57	0,60
oracle-DB	0,64	0,58	0,61
oracle+oracle-DB	0,70	0,64	**0,67**
char3+oracle	0,70	0,62	0,66
char3+oracle+word	0,69	0,56	0,62
all	0,73	0,64	**0,68**

Table 5.11 : Ensemble of base models via stack generalization. Models in italics are provided as references.

5.3.4 Multiple Units: German, Spanish, Czech, Catalan, Finnish

Here, we focus on post integration techniques to test whether our results from previous section hold for other languages as well. It should be noted that, we only vary the input representation in subword models to test whether subwords may help each other. Hence all base models are composed with bi-LSTMs and trained with same hyperparameters.

The integration results for averaging and SG are given in Table 5.12. According to this table, combining char and char3 improves results at least by 1 point for all languages independent from the ensembling technique. In contrast to our findings for char3 and oracle integration, this combination does not improve on the results of oracle for

some languages. Interestingly, it does provide great benefit to German, Spanish and Catalan while it does have no effect on Finnish and Czech, the ones with the richest morphology. It may suggest that characters do not capture any information that is not already in oracle models.

The type of ensembling technique did not have a big impact on the results. Although SG performed slightly better than averaging for Catalan and Finnish while combining char and char3; and char, char3 and oracle for German.

		Average			SG		
		P	**R**	**F1**	**P**	**R**	**F1**
German	*char3*	0,66	0,62	0,64	0,66	0,62	0,64
	char+char3	0,68	0,64	0,66	0,69	0,64	0,66
	oracle	0,68	0,64	0,66	0,68	0,64	**0,66**
	char+char3+oracle	0,72	0,66	**0,69**	0,75	0,68	**0,71**
Spanish	*char3*	0,71	0,66	0,68	0,71	0,66	0,68
	char+char3	0,72	0,67	0,69	0,72	0,67	0,69
	oracle	0,68	0,69	0,69	0,68	0,69	**0,69**
	char+char3+oracle	0,74	0,70	0,72	0,74	0,70	**0,72**
Czech	*char3*	0,80	0,74	0,77	0,80	0,74	0,77
	char+char3	0,83	0,74	0,78	0,82	0,74	0,78
	oracle	0,84	0,78	0,81	0,84	0,78	**0,81**
	char+char3+oracle	0,86	0,76	0,81	0,86	0,77	**0,81**
Catalan	*char3*	0,72	0,66	0,69	0,72	0,66	0,69
	char+char3	0,75	0,64	**0,69**	0,74	0,67	**0,71**
	oracle	0,71	0,69	0,70	0,71	0,69	**0,70**
	char+char3+oracle	0,76	0,70	0,73	0,76	0,71	**0,73**
Finnish	*char3*	0,70	0,64	0,67	0,70	0,64	0,67
	char+char3	0,66	0,65	**0,66**	0,75	0,64	**0,69**
	oracle	0,74	0,71	0,72	0,74	0,71	**0,72**
	char+char3+oracle	0,78	0,67	0,72	0,78	0,67	**0,72**

Table 5.12 : Post integration results for other languages

5.4 Analysis of Experiments

We analyze the label errors made by different subunits (single subwords) in order to understand the biases of each base model. Afterwards, we extend our analysis to other languages to compare the biases of base models across languages. Finally we compare results of neural SRL with statistical SRL to gain a deeper understanding of the strengths and weaknesses of the neural model.

5.4.1 Label Error Analysis

Turkish

In Table 5.13, P, R and F1 scores for each label (sorted by their count) is given. Table shows that, **oracle-DB**, the method that gives the best labeled and unlabeled F1 scores, is not the best for each label. However, there seems to be a nice interplay between oracle and oracle-DB models. For instance, while oracle surpasses the performance of oracle-DB on classification of A2, A3, AM-COM and AM-DIR labels, the opposite holds for A-A, A0 and AM-TMP roles.

We immediately observe that char3 model's performance is better than oracle-DB and close to oracle in A2 and A4 and AM-ADV classification and for the temporary arguments AM-EXT, AM-CAU and AM-GOL, char outperforms any morphological model. However, for labeling A-A and AM-TWO, information acquired only from characters seems to be insufficient.

Other Languages

Similar analysis has been performed for Finnish (see Table A.4), German (see Table A.5), Spanish (see Table A.6), Catalan (see Table A.7) and Czech (see Table A.8). Similar to our previous finding on Turkish, we have observed that different subunits achieve better F1 scores on different semantic role labels.

Generally speaking, we have recognized a few repeating patterns for character based models across languages. They achieved scores higher than oracle models for predicting **causes** (AM-CAU) both for Turkish and Finnish; **extents** and **destinations** for Turkish, Spanish and Catalan; **initial** (ein) and **final states** (efi) for Spanish and Catalan. Since argument labels for German and Czech use a different scheme, it is not possible to perform a detailed analysis on temporary arguments for which defnitions are not clear. In addition, we have searched for a failure pattern for character models. Interestingly, for all languages that we have evaluated our model, their performance was poor for labeling A0 (agent, experiencer, causer)[2].

[2]A0 label has more consistency among languages. German, Turkish and Finnish A0s serve the default PropBank purpose (agent, experiencer, causer). Spanish and Catalan define fine-grained A0s: arg0-agt for agent, arg0-cau for causer. There was a performance drop for both roles (bigger on arg0-cau). ACT is the corresponding role for A0 in Czech corpus.

		char			char3			char5			oracle			oracle-DB			bpe			morfessor		
Label	**n**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
A1	1209	0,72	**0,75**	0,73	0,74	0,71	0,73	0,66	0,63	0,64	0,75	0,74	0,74	**0,79**	0,73	**0,76**	0,67	0,63	0,65	0,63	0,64	0,63
A0	567	0,49	0,42	0,45	0,51	0,38	0,43	0,45	0,34	0,39	0,56	0,47	0,51	**0,60**	**0,57**	**0,59**	0,44	0,32	0,37	0,47	0,34	0,39
AM-TMP	218	0,51	0,61	0,56	0,58	0,56	0,57	0,52	0,44	0,48	0,59	0,55	0,57	**0,62**	**0,62**	**0,62**	0,49	0,42	0,45	0,52	0,45	0,49
AM-MNR	208	0,56	0,53	0,54	0,51	0,50	0,51	0,44	0,42	0,43	0,52	**0,58**	**0,55**	**0,59**	0,50	0,54	0,49	0,43	0,46	0,50	0,33	0,40
A2	201	0,49	0,36	0,42	0,56	0,45	0,50	0,47	0,39	0,43	**0,56**	**0,47**	**0,51**	0,35	0,38	0,37	0,41	0,30	0,35	0,37	0,20	0,26
AM-LOC	138	0,47	0,51	0,49	0,50	0,48	0,49	0,44	0,33	0,38	0,52	0,47	0,49	0,50	**0,56**	**0,53**	0,41	0,31	0,35	**0,54**	0,25	0,34
AM-LVB	100	**0,84**	0,81	0,83	0,80	0,82	0,81	0,83	0,72	0,77	0,80	**0,88**	**0,84**	0,78	0,83	0,81	0,71	0,76	0,73	0,73	0,65	0,69
A4	74	**0,68**	**0,65**	**0,66**	0,68	0,57	0,62	0,50	0,50	0,50	0,68	0,64	0,66	0,51	0,53	0,52	0,46	0,49	0,47	0,55	0,36	0,44
AM-EXT	67	**0,64**	0,43	**0,52**	0,50	0,46	0,48	0,57	0,37	0,45	0,50	**0,49**	0,50	0,61	0,34	0,44	0,45	0,43	0,44	0,41	0,39	0,40
AM-CAU	62	0,44	**0,29**	**0,35**	0,42	0,29	0,34	0,38	0,24	0,30	**0,45**	0,27	0,34	0,38	0,26	0,31	0,42	0,27	0,33	0,40	0,23	0,29
AM-GOL	57	**0,38**	**0,32**	**0,34**	0,31	0,23	0,26	0,27	0,14	0,18	0,29	0,26	0,28	0,26	0,16	0,20	0,19	0,11	0,13	0,20	0,16	0,17
A3	46	0,54	0,33	0,41	0,43	0,41	0,42	0,53	0,35	0,42	**0,69**	**0,52**	**0,59**	0,42	0,48	0,44	0,57	0,37	0,45	0,47	0,20	0,28
AM-ADV	38	0,38	0,26	0,31	**0,40**	**0,32**	**0,35**	0,34	0,26	0,30	0,38	0,32	0,34	0,35	0,24	0,28	0,25	0,16	0,19	0,26	0,16	0,20
AM-PRD	34	0,43	0,18	0,25	**0,56**	0,15	0,23	0,45	0,15	0,22	0,50	0,21	0,29	0,53	**0,26**	**0,35**	0,14	0,06	0,08	0,25	0,09	0,13
AM-DIS	27	0,33	**0,11**	**0,17**	0,33	0,07	0,12	0,40	0,07	0,12	0,25	0,04	0,06	**1**	0,04	0,07	0,27	0,11	0,16	0	0	0
A-A	23	0	0	0	0,25	0,04	0,07	0	0	0	0,25	0,09	0,13	**0,57**	**0,35**	**0,43**	0	0	0	0	0	0
AM-INS	22	0,28	0,23	0,25	0,23	0,23	0,23	0,22	0,18	0,20	0,30	0,32	0,31	0,32	**0,41**	**0,36**	**0,40**	0,27	0,32	0,38	0,23	0,29
C-A1	20	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
AM-TWO	13	**0,57**	0,31	0,40	0,30	0,23	0,26	0,33	0,23	0,27	0,50	0,38	0,43	0,47	**0,54**	**0,50**	0,50	0,15	0,24	0,50	0,23	0,32
AM-COM	13	0,22	0,15	0,18	0,20	0,08	0,11	0,33	0,08	0,12	**0,33**	**0,15**	**0,21**	0,20	0,08	0,11	0	0	0	0,20	0,08	0,11
AM-DIR	12	0,27	0,33	0,30	0,30	0,25	0,27	0,40	0,17	0,24	0,44	**0,33**	**0,38**	**1**	0	0	0,20	0,08	0,12	0,12	0,08	0,10
AM-NEG	11	**1**	**0,64**	**0,78**	1	0,64	0,78	0,88	0,64	0,74	1	0,64	0,78	1	0,55	0,71	0,83	0,45	0,59	0,75	0,55	0,63

Table 5.13 : **Turkish:** Label errors made by each unit (bi-LSTM) composition. Best precision (P), recall (R) and F1 scores are given in bold.

5.4.2 Comparison with statistical model

We compare the best neural model (stacked ensemble) with statistical model described in previous chapter. In Fig. 5.8, differences between precision, recall and F1 scores of semantic role labels classified by statistical and neural models are shown. We limit the labels to ones that occur more than 50 times for a meaningful analysis. According to the figure, for almost all labels statistical model surpasses the performance of the neural model. The value for recall differences are bigger than the precision ones. It shows that, the neural model is weak at identifying the arguments rather than assigning labels. Following our previous analysis, A0 role is the most difficult semantic role to be classified without high level syntactic features such as dependency labels.

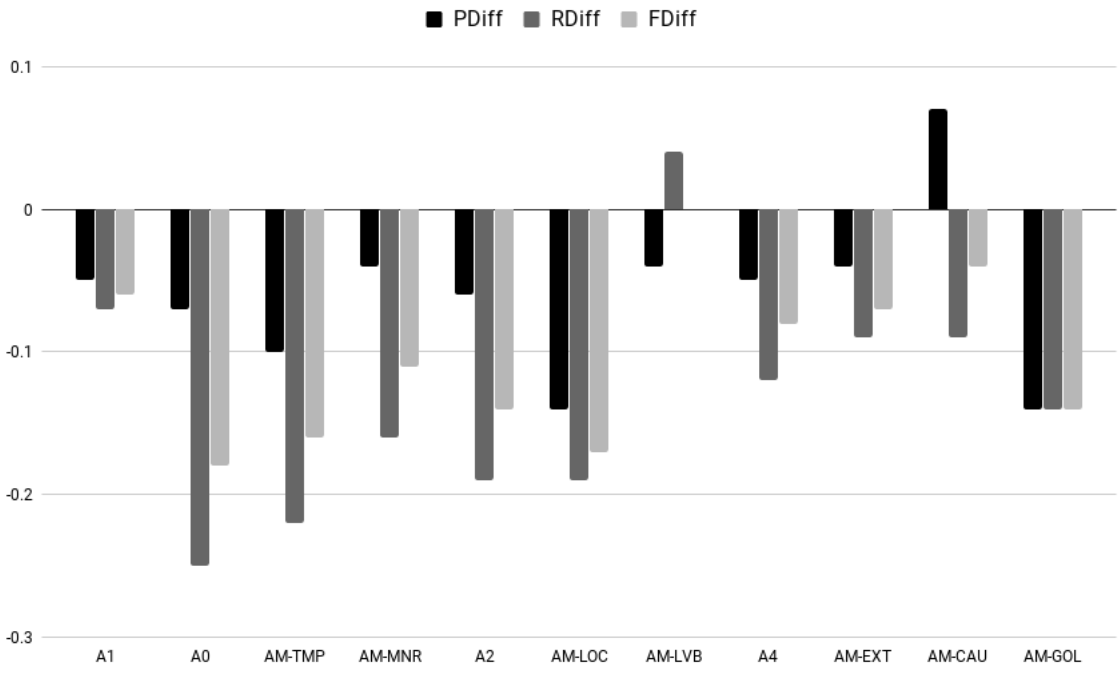


Figure 5.8 : Precision, Recall and F1 differences per role

Next, we compare their overall argument labeling performances with respect to the sentence length. The purpose of this analysis is to understand whether one method is superior to the other in short/long sentences. The result is given in Fig. 5.9. The overarall difference between errors has the mean 0,10 and the standard deviation of 0,11. The gap between errors has the mean 0,14 for the second and third quarter,

while around 0,04 for the first and the last quarter, which makes it hard to draw any conclusions.

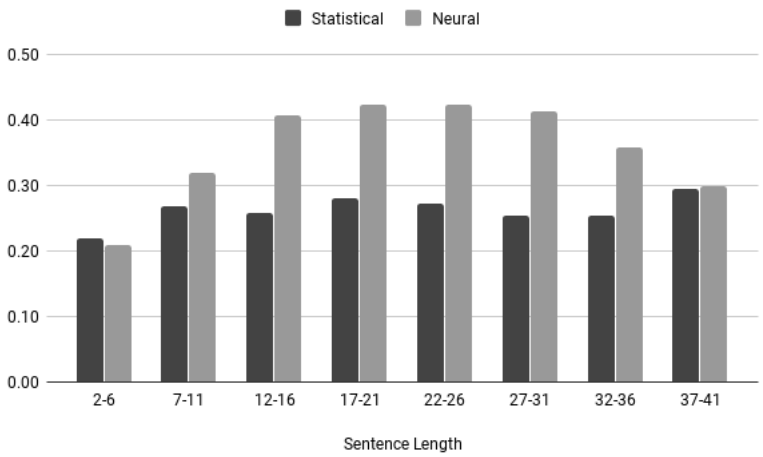


Figure 5.9 : Comparison of statistical and neural models wrt sentence length

Finally, we evaluate their performance on handling long range dependencies as shown in Fig. 5.10. Performance of both models go down as the argument's distance to verb increases until the distance is 9. Then neural model stabilizes while statistical model makes more mistakes. Still, statistical model outperforms neural model for arguments relatively closer to its predicate.

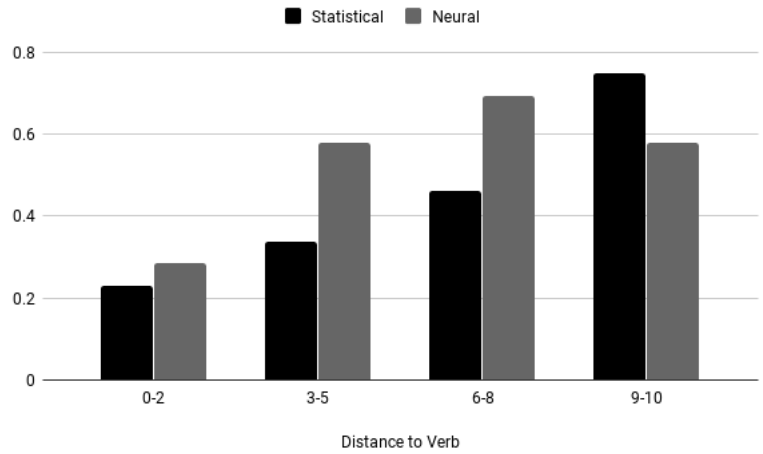


Figure 5.10 : Mistakes by statistical and neural models wrt long range dependencies

5.4.2.1 Weaknesses and Strengths

Argument Identification:

We have compared the argument identification results of two systems (best of

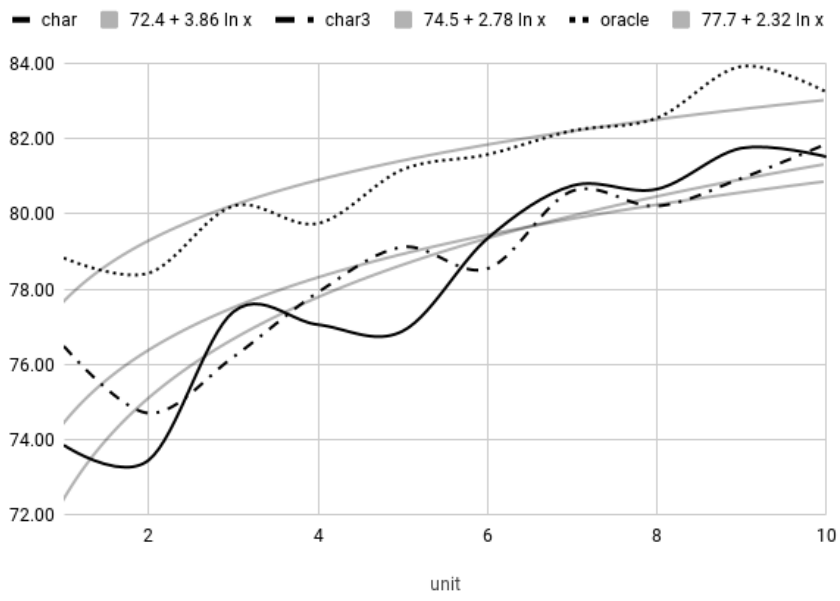


Figure 5.11 : PD accuracy versus datasize

both). According to those results, the biggest weakness of the neural model is its low capability of identifying arguments. However once arguments are identified, classification is performed with better accuracy. For instance statistical system had the performance of P=0,97, R=0,95 and F1=0,96 while the best ensemble model achieved only P=0,90, R=0,78 and F1=0,83. The bottleneck of the system seems to be identifying the *actual* arguments.

5.4.3 Dataset Experiment

The relationship between PD accuracy and the size of the training data is given in Fig. 5.11. The equations for the best fitting logarithmic growth for char, char3 and oracle are given on top of the figure. These results suggest that, all character models can benefit more from more training data than oracle models. Therefore, we expect character models perform the same with or better than oracle models in presence of more data.

5.5 Summary

This chapter discussed a neural Turkish SRL system based on long-short term memory units. We have introduced the OOV problem that causes extensive performance decrease on syntax-agnostic end-to-end neural methods which only rely only on word

information. To address rare words, we have proposed using smaller units referred to as subwords that are based on:

- characters and character sequences
- morphemes with prior language specific frequency knowledge
- morphological analysis

Then we compose them via functions with varying complexities to generate a word embedding. Available subword composition techniques did not make any distinctions between morphology types. We have introduced a linguistically motivated composition technique that distinguishes derivational morphology from inflectional one and reported higher F1 scores on argument labeling. We have systematically analyzed the effect of subword and composition types. We concluded that character based models with bi-LSTM composition (specifically character trigrams) perform almost as good as morphological ones for the languages German, Spanish and Catalan; whereas at least 3pp drop has been observed on F1 scores for morphologically rich languages Czech, Finnish and Turkish. Next, we designed experiments to test the actual hypothesis: *Subword units provide complementary information for argument labeling task*. We have integrated subwords at word encoding and later at ensembling stage via stack generalization and simple product-of-experts method. We have observed that post integration performs better and is less computationally demanding; in addition a simple product-of-experts moslty yielded to the same scores as stack generalization. Independent from the language and the ensembling technique, char and char trigram combination always improved the scores. However we have seen mixed results for combining character information with morphology. Although the scores have improved for Turkish, German, Spanish and Catalan, it had no effect on Finnish and Czech. It suggests that characters do not capture any information that is not already in oracle models for those languages.

6. Conclusion and Future Work

Semantic role labeling especially for non-English languages suffers from the lack of necessary resources. To fill this gap, we have built the first SRL resource for Turkish: Turkish PropBank. It is constructed on top of Itu-Metu-Sabancı TreeBank (IMST) that contains 5635 sentences, is morphologically analysed, POS tagged and manually labeled with shallow and deep dependencies. We have discussed the complete construction workflow: framing, corwdsourcing of verb sense disambiguation and semantic role annotation.

First, we have introduced Turkish-specific challenges for SRL task: the immense syntactic variation and infinite word lexicon problem caused by derivational morphemes. We have proposed and demonstrated the feasibility of our exploiting framesets of root verbs approach. We showed that this approach enables us to abstract farther away from syntax and increase self-consistency of Turkish PropBank. We presented the framing guidelines for distinguishing sense and argument numbers including exceptional cases caused by valency changing morphemes, light verbs, multiword expressions and nominal verbs; and released a verb lexicon framed with PropBank annotation scheme.

We have described verb sense disambiguation and semantic role annotation of arguments in the treebank with the help of crowds. Our quality control mechanism based on the idea of preparing of expert labeled test questions is discussed. We have continuously removed under-performers, trained crowdworkers and given them real-time feedback with the help of this mechanism. We demonstrated the feasibility of our approach on annotation examples of verbal nominals, nominal verbs and copulas. We have evaluated the annotation quality by means of various inter-annotator metrics such as kappa scored that measure agreements among crowdworkers and experts. We have discussed the possible causes of disagreements and how to address them. Furthermore, we discuss the adjudication, handling of continuous arguments and copulas after crowdsourcing. Universal dependency compliant treebank IMST-UD has

been provided with semantic annotations by automatically aligning the semantic layer of IMST.

We have presented a logistic regression classifier based framework for automatically extracting semantic roles. Discrete features that require outputs of external NLP tools, such as dependency paths and postags, have been evaluated. To address shortcomings of discrete features, we have introduced continuous features based on pretrained word embeddings. We showed that morphosemantic features are important for a high performing SRL task. Size of training data has a larger impact on predicate sense disambiguation than on argument labeling and an acceptable labeling system can be achieved with almost 60% of training data in the presence of well designed features. We have discussed that it is not possible to achieve high scores with a statistical SRL system without high-level features. Our experiments showed that although continuous features can not be used as a substitute for other levels of features, they provide improvement over scores when replaced with postag information. It suggests that these features contain syntactic information. We increased the performance of the first system by incorporating continuous features. This can be interpreted as continuous features enable us to model complex interactions between information levels. We have achieved an F1 score of 79.84. We report that F1 scores are well within the expected range considering the performance range (76.30 -85.63) of adopted system. We make all annotated resources, as well as the predicate lexicon containing semantic frames, and source code freely available to enable the development of high-performance Turkish SRL systems and high level language understanding studies. http://turkishpropbank.github.io/.

One of the major drawbacks of this resource is its small size, which we plan to increase via running Turkish SRL on text and have it corrected by crowdworkers or semi-automatic methods in near future. Another future work is to build a nominal bank for Turkish to be able to annotate argument structures of nominals. Finally we plan to optimize features of the UD scheme for a better F1 score on Turkish SRL.

REFERENCES

[1] **Woods, W.A.** (1967). Semantics for a question-answering system, *Ph.D. thesis*, Harvard University.

[2] **Quillian, M.** (1968). Semantic Information Processing.

[3] **Fillmore, C.J.** (1967). The case for case.

[4] **Baker, C.F.**, **Fillmore, C.J. and Lowe, J.B.** (1998). The berkeley FrameNet project, *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, Association for Computational Linguistics, pp.86–90.

[5] **Schuler, K.K.** (2005). VerbNet: A broad-coverage, comprehensive verb lexicon.

[6] **Palmer, M.**, **Gildea, D. and Kingsbury, P.** (2005). The proposition bank: An annotated corpus of semantic roles, *Computational linguistics*, *31*(1), 71–106.

[7] **Banarescu, L.**, **Bonial, C.**, **Cai, S.**, **Georgescu, M.**, **Griffitt, K.**, **Hermjakob, U.**, **Knight, K.**, **Koehn, P.**, **Palmer, M. and Schneider, N.** (2012). Abstract meaning representation (AMR) 1.0 specification, *Parsing on Freebase from Question-Answer Pairs." In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. Seattle: ACL*, pp.1533–1544.

[8] **Vanderwende, L.**, **Menezes, A. and Quirk, C.** (2015). An AMR parser for English, French, German, Spanish and Japanese and a new AMR-annotated corpus.

[9] **May, J.** (2016). SemEval-2016 Task 8: Meaning Representation Parsing, *Proceedings of SemEval*, 1063–1073.

[10] **Giuglea, A.M. and Moschitti, A.** (2006). Semantic role labeling via framenet, verbnet and propbank, *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, pp.929–936.

[11] **Palmer, M.**, **Bhatt, R.**, **Narasimhan, B.**, **Rambow, O.**, **Sharma, D.M. and Xia, F.** (2009). Hindi Syntax: Annotating Dependency, Lexical Predicate-Argument Structure, and Phrase Structure, *Proceedings of the 7th International Conference on Natural Language Processing, ICON'09*, 261—-268.

[12] **Xue, N. and Palmer, M.** (2008). Adding semantic roles to the Chinese Treebank, *Natural Language Engineering*, *15*(1), 143.

[13] **Zaghouani, W.**, **Diab, M.**, **Mansouri, A.**, **Pradhan, S. and Palmer, M.** (2010). The revised Arabic PropBank, *10 Proceedings of the Fourth Linguistic Annotation Workshop*, 222–226.

[14] **Haverinen, K.**, **Kanerva, J.**, **Kohonen, S.**, **Missila, A.**, **Ojala, S.**, **Viljanen, T.**, **Laippala, V. and Ginter, F.** (2015). The Finnish Proposition Bank, *Language Resources and Evaluation*, *49*(4), 907–926.

[15] **Duran, M.S. and Aluísio, S.M.** (2012). Propbank-Br: a Brazilian Treebank annotated with semantic role labels, *LREC*.

[16] **Carreras, X. and Màrquez, L.** (2004). Introduction to the CoNLL-2004 shared task: Semantic role labeling, *Proceedings of the Ninth Conference on Computational Natural Language Learning*, Association for Computational Linguistics.

[17] **Carreras, X. and Màrquez, L.** (2005). Introduction to the CoNLL-2005 shared task: Semantic role labeling, *Proceedings of the Ninth Conference on Computational Natural Language Learning*, Association for Computational Linguistics, pp.152–164.

[18] **Surdeanu, M.**, **Johansson, R.**, **Meyers, A.**, **Màrquez, L. and Nivre, J.** (2008). The CoNLL-2008 shared task on joint parsing of syntactic and semantic dependencies, *Proceedings of the Twelfth Conference on Computational Natural Language Learning*, Association for Computational Linguistics, pp.159–177.

[19] **Hajič, J.**, **Ciaramita, M.**, **Johansson, R.**, **Kawahara, D.**, **Martí, M.A.**, **Màrquez, L.**, **Meyers, A.**, **Nivre, J.**, **Padó, S.**, **Štěpánek, J.**, **Straňák, P.**, **Surdeanu, M.**, **Xue, N. and Zhang, Y.** (2009). The CoNLL-2009 Shared Task: Syntactic and Semantic Dependencies in Multiple Languages, *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*, CoNLL '09, Association for Computational Linguistics, Stroudsburg, PA, USA, pp.1–18.

[20] **Meyers, A.**, **Reeves, R.**, **Macleod, C.**, **Szekely, R.**, **Zielinska, V.**, **Young, B. and Grishman, R.** (2004). The NomBank project: An interim report, *HLT-NAACL 2004 workshop: Frontiers in corpus annotation*, volume 24, p. 31.

[21] **Oepen, S.**, **Kuhlmann, M.**, **Miyao, Y.**, **Zeman, D.**, **Flickinger, D.**, **Hajic, J.**, **Ivanova, A. and Zhang, Y.** (2014). SemEval 2014 Task 8: Broad-coverage semantic dependency parsing, *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pp.63–72.

[22] **Oepen, S.**, **Kuhlmann, M.**, **Miyao, Y.**, **Zeman, D.**, **Cinková, S.**, **Flickinger, D.**, **Hajic, J. and Uresová, Z.** (2015). SemEval 2015 Task 18: Broad-Coverage Semantic Dependency Parsing, *SemEval@NAACL-HLT*.

[23] **Christensen, J.**, **Soderland, S.**, **Etzioni, O.** *et al.* (2010). Semantic role labeling for open information extraction, *Proceedings of the NAACL HLT 2010 First International Workshop on Formalisms and Methodology for Learning by Reading*, Association for Computational Linguistics, pp.52–60.

[24] **Hung, S.H.**, **Lin, C.H. and Hong, J.S.** (2010). Web mining for event-based commonsense knowledge using lexico-syntactic pattern matching and semantic role labeling, *Expert Systems with Applications*, *37*(1), 341–347.

[25] **Exner, P. and Nugues, P.** (2011). Using semantic role labeling to extract events from Wikipedia, *Proceedings of the Workshop on Detection, Representation, and Exploitation of Events in the Semantic Web (DeRiVE 2011). Workshop in conjunction with the 10th International Semantic Web Conference*, pp.23–24.

[26] **Shen, D. and Lapata, M.** (2007). Using Semantic Roles to Improve Question Answering., *EMNLP-CoNLL*, pp.12–21.

[27] **Kaisser, M. and Webber, B.** (2007). Question answering based on semantic roles, *Proceedings of the Workshop on Deep Linguistic Processing*, Association for Computational Linguistics, pp.41–48.

[28] **Sammons, M.**, **Vydiswaran, V.V.**, **Vieira, T.**, **Johri, N.**, **Chang, M.W.**, **Goldwasser, D.**, **Srikumar, V.**, **Kundu, G.**, **Tu, Y.**, **Small, K.** *et al.* (2009). Relation alignment for textual entailment recognition, *Text Analysis Conference (TAC)*.

[29] **Wu, D. and Fung, P.** (2009). Can semantic role labeling improve SMT, *Proceedings of the 13th Annual Conference of the EAMT*, pp.218–225.

[30] **Liu, D. and Gildea, D.** (2010). Semantic role features for machine translation, *Proceedings of the 23rd International Conference on Computational Linguistics*, Association for Computational Linguistics, pp.716–724.

[31] **Lo, C.k.**, **Addanki, K.**, **Saers, M. and Wu, D.** (2013). Improving machine translation by training against an automatic semantic frame based evaluation metric., *ACL*, pp.375–381.

[32] **Gao, Q. and Vogel, S.** (2011). Corpus expansion for statistical machine translation with semantic role label substitution rules, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, Association for Computational Linguistics, pp.294–298.

[33] **Akbik, A.**, **chiticariu, l.**, **Danilevsky, M.**, **Li, Y.**, **Vaithyanathan, S. and Zhu, H.** (2015). Generating High Quality Proposition Banks for Multilingual Semantic Role Labeling, *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Beijing, China, pp.397–407.

[34] **Oflazer, K. and El-Kahlout, I.D.** (2007). Exploring different representational units in English-to-Turkish statistical machine translation, *Proceedings of the Second Workshop on Statistical Machine Translation*, Association for Computational Linguistics, pp.25–32.

[35] **Agirre, E.**, **Aldezabal, I.**, **Etxeberria, J. and Pociello, E.** (2006). A preliminary study for building the Basque PropBank, *Proceedings of the 5th International Conference on Language Resources and Evaluations (LREC)*.

[36] **Aldezabal, I.**, **Aranzabe, M.J.**, **de Ilarraza Sánchez, A.D. and Estarrona, A.** (2010). Building the Basque PropBank., *LREC*.

[37] **Hawwari, A.**, **Zaghouani, W.**, **O'Gorman, T.**, **Badran, A. and Diab, M.** (2013). Building a lexical semantic resource for Arabic morphological Patterns, *Communications, Signal Processing, and their Applications (ICCSPA), 2013 1st International Conference on*, IEEE, pp.1–6.

[38] **Fellbaum, C.**, **Osherson, A. and Clark, P.E.** (2007). Putting semantics into WordNet's" morphosemantic" links, *Language and Technology Conference*, Springer, pp.350–358.

[39] **Bilgin, O.**, **Çetinoğlu, Ö. and Oflazer, K.** (2004). Building a wordnet for Turkish, *Romanian Journal of Information Science and Technology*, *7*(1-2), 163–172.

[40] **Mititelu, V.B.** (2012). Adding Morpho-semantic Relations to the Romanian Wordnet., *LREC*, pp.2596–2601.

[41] **Babko-Malaya, O.** (2005). Guidelines for Propbank framers, *Unpublished manual, September*.

[42] **Choi, J.D.**, **Bonial, C. and Palmer, M.** (2010). Propbank Frameset Annotation Guidelines Using a Dedicated Editor, Cornerstone., *LREC*.

[43] **Uçar, A.** (2010). Light Verb Constructions in Turkish Dictionaries Are they submeanings of polysemous verbs?, *Dil ve Edebiyat Dergisi*, *7*(1).

[44] **Hwang, J.D.**, **Bhatia, A.**, **Bonial, C.**, **Mansouri, A.**, **Vaidya, A.**, **Xue, N. and Palmer, M.** (2010). Propbank annotation of multilingual light verb constructions, *Proceedings of the Fourth Linguistic Annotation Workshop*, Association for Computational Linguistics, pp.82–90.

[45] **Butt, M.** (2010). The light verb jungle: still hacking away, *Complex predicates in cross-linguistic perspective*, 48–78.

[46] **Oflazer, K.** (2014). Turkish and its challenges for language processing, *Language Resources and Evaluation*, *48*(4), 639–653.

[47] **Sahin, G.G. and Adalı, E.** (2014). Using Morphosemantic Information in Construction of a Pilot Lexical Semantic Resource for Turkish, *Workshop on Lexical and Grammatical Resources for Language Processing*, p. 46.

[48] **Hengirmen, M.** (2006). *Türkçe temel dilbilgisi*, Engin Yayınevi.

[49] **Sahin, G.G.**, (2016). Framing of Verbs for Turkish PropBank, In Proceedings of 1st International Conference on Turkic Computational Linguistics, TurCLing.

[50] **Mohammad, S.M. and Turney, P.D.** (2013). Crowdsourcing a word–emotion association lexicon, *Computational Intelligence*, *29*(3), 436–465.

[51] **Basile, V.**, **Bos, J.**, **Evang, K. and Venhuizen, N.** (2012). Developing a large semantically annotated corpus., *LREC*, volume 12, pp.3196–3200.

[52] **Mellebeek, B.**, **Benavent, F.**, **Grivolla, J.**, **Codina, J.**, **Costa-Jussa, M.R. and Banchs, R.** (2010). Opinion mining of spanish customer comments with non-expert annotations on mechanical turk, *Proceedings of the NAACL HLT 2010 workshop on Creating speech and language data with Amazon's mechanical turk*, Association for Computational Linguistics, pp.114–121.

[53] **Zaidan, O.F. and Callison-Burch, C.** (2011). Crowdsourcing translation: Professional quality from non-professionals, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, Association for Computational Linguistics, pp.1220–1229.

[54] **Madnani, N.**, **Tetreault, J.**, **Chodorow, M. and Rozovskaya, A.** (2011). They can help: using crowdsourcing to improve the evaluation of grammatical error detection systems, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, Association for Computational Linguistics, pp.508–513.

[55] **Zhai, H.**, **Lingren, T.**, **Deleger, L.**, **Li, Q.**, **Kaiser, M.**, **Stoutenborough, L. and Solti, I.** (2013). Web 2.0-based crowdsourcing for high-quality gold standard development in clinical natural language processing, *Journal of medical Internet research*, *15*(4).

[56] **Hong, J. and Baker, C.F.** (2011). How good is the crowd at real WSD?, *Proceedings of the 5th linguistic annotation workshop*, Association for Computational Linguistics, pp.30–37.

[57] **Sahin, G.G.** (2016). Verb Sense Annotation For Turkish PropBank via Crowdsourcing, *Proceedings of 17th International Conference on Intelligent Text Processing and Computational Linguistics*, CICLING 2016.

[58] **Snow, R.**, **O'Connor, B.**, **Jurafsky, D. and Ng, A.Y.** (2008). Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks, *Proceedings of the conference on empirical methods in natural language processing*, Association for Computational Linguistics, pp.254–263.

[59] **Callison-Burch, C.**, **Ungar, L. and Pavlick, E.** (2015). Crowdsourcing for NLP, *Proceedings of NAACL 2015*, North America Association for Computational Linguistics.

[60] **Sabou, M.**, **Bontcheva, K.**, **Derczynski, L. and Scharl, A.** (2014). Corpus annotation through crowdsourcing: Towards best practice guidelines, *Proceedings of LREC*.

[61] **Feizabadi, P.S. and Padó, S.** (2014). Crowdsourcing Annotation of Non-Local Semantic Roles., *EACL*, pp.226–230.

[62] **Fossati, M.**, **Giuliano, C. and Tonelli, S.** (2013). Outsourcing FrameNet to the Crowd., *ACL (2)*, pp.742–747.

[63] **Fossati, M.**, **Tonelli, S. and Giuliano, C.** (2013). Frame Semantics Annotation Made Easy with DBpedia, *Crowdsourcing, the, Semantic, Web*.

[64] **Chang, N.**, **Paritosh, P.**, **Huynh, D. and Baker, C.F.** (2015). Scaling Semantic Frame Annotation, *The 9th Linguistic Annotation Workshop held in conjuncion with NAACL 2015*, p. 1.

[65] **Reisinger, D.**, **Rudinger, R.**, **Ferraro, F.**, **Harman, C.**, **Rawlins, K. and Van Durme, B.** (2015). Semantic Proto-Roles, *Transactions of the Association for Computational Linguistics*, *3*, 475–488.

[66] **He, L.**, **Lewis, M. and Zettlemoyer, L.** (2015). Question-Answer Driven Semantic Role Labeling: Using Natural Language to Annotate Natural Language, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Lisbon, Portugal, pp.643–653.

[67] **Rim, K.**, (2015), Event-Participant Linking-A Crowdsourcing Approach to Semantic Role Labeling.

[68] **Pavlick, E.**, **Post, M.**, **Irvine, A.**, **Kachaev, D. and Callison-Burch, C.** (2014). The language demographics of amazon mechanical turk, *Transactions of the Association for Computational Linguistics*, *2*, 79–92.

[69] **Irvine, A. and Klementiev, A.** (2010). Using Mechanical Turk to annotate lexicons for less commonly used languages, *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, Association for Computational Linguistics, pp.108–113.

[70] **Sulubacak, U.**, **Pamay, T. and Eryigit, G.**, (2016). IMST: A Revisited Turkish Dependency Treebank, In Proceedings of 1st International Conference on Turkic Computational Linguistics, TurCLing.

[71] **Oflazer, K.**, **Say, B.**, **Hakkani-Tür, D.Z. and Tür, G.**, (2003). Building a Turkish treebank, Treebanks, Springer, pp.261–277.

[72] **Schuster, S. and Manning, C.D.** (2016). Enhanced english universal dependencies: An improved representation for natural language understanding tasks, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*.

[73] **Lewis, G.L.** (1985). *Turkish grammar*, Oxford University Press, USA.

[74] **Xue, N. and Palmer, M.** (2009). Adding semantic roles to the Chinese Treebank, *Natural Language Engineering*, *15*(01), 143–172.

[75] **Fleiss, J.L.** (1971). Measuring nominal scale agreement among many raters., *Psychological bulletin*, *76*(5), 378.

[76] **Sulubacak, U.**, **Gökırmak, M. and Eryiğit, G.** (2016). Universal Dependencies for Turkish, *COLING*.

[77] **Saffran, J.R.**, **Johnson, E.K.**, **Aslin, R.N. and Newport, E.L.** (1999). Statistical learning of tone sequences by human infants and adults, *Cognition*, *70*(1), 27–52.

[78] **Saffran, J.R.** (2003). Statistical language learning mechanisms and constraints, *Current directions in psychological science*, *12*(4), 110–114.

[79] **Gildea, D.** (2002). Automatic labeling of semantic roles, *Computational Linguistics*, *28*(3), 245–288.

[80] **Deschacht, K. and Moens, M.F.** (2009). Semi-supervised semantic role labeling using the latent words language model, *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, Association for Computational Linguistics, pp.21–29.

[81] **Fürstenau, H. and Lapata, M.** (2009). Graph alignment for semi-supervised semantic role labeling, *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing Volume 1 - EMNLP '09*, (August), 11–20.

[82] **Padó, S. and Lapata, M.** (2009). Cross-lingual annotation projection for semantic roles, *Journal of Artificial Intelligence Research*, *36*(1), 307–340.

[83] **Kozhevnikov, M. and Titov, I.** (2013). Cross-lingual Transfer of Semantic Role Labeling Models., *ACL (1)*, pp.1190–1200.

[84] **Titov, I. and Klementiev, A.** (2011). A Bayesian model for unsupervised semantic parsing, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, Association for Computational Linguistics, pp.1445–1455.

[85] **Toutanova, K.**, **Haghighi, A. and Manning, C.D.** (2005). Joint learning improves semantic role labeling, *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, Association for Computational Linguistics, pp.589–596.

[86] **Björkelund, A.**, **Hafdell, L. and Nugues, P.** (2009). Multilingual semantic role labeling, *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*, Association for Computational Linguistics, pp.43–48.

[87] **Roth, M. and Woodsend, K.** (2014). Composition of Word Representations Improves Semantic Role Labelling., *EMNLP*, pp.407–413.

[88] **Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K. and Kuksa, P.** (2011). Natural Language Processing (almost) from Scratch, *Journal of Machine Learning Research*, *12*, 2461–2505.

[89] **FitzGerald, N., Täckström, O., Ganchev, K. and Das, D.** (2015). Semantic Role Labeling with Neural Network Factors., *EMNLP*, pp.960–970.

[90] **Roth, M. and Lapata, M.** (2016). Neural semantic role labeling with dependency path embeddings, *arXiv preprint arXiv:1605.07515*.

[91] **Zhou, J. and Xu, W.** (2015). End-to-end learning of semantic role labeling using recurrent neural networks., *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp.1127–1137.

[92] **Wang, Z., Jiang, T., Chang, B. and Sui, Z.** (2015). Chinese Semantic Role Labeling with Bidirectional Recurrent Neural Networks., *EMNLP*, pp.1626–1631.

[93] **Marcheggiani, D., Frolov, A. and Titov, I.** (2017). A Simple and Accurate Syntax-Agnostic Neural Model for Dependency-based Semantic Role Labeling, *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, Association for Computational Linguistics, Vancouver, Canada, pp.411–420, http://aclweb.org/anthology/K17-1041.

[94] **Marcheggiani, D. and Titov, I.** (2017). Encoding Sentences with Graph Convolutional Networks for Semantic Role Labeling, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Copenhagen, Denmark, pp.1507–1516, https://www.aclweb.org/anthology/D17-1159.

[95] **He, L., Lee, K., Lewis, M. and Zettlemoyer, L.** (2017). Deep semantic role labeling: What works and what's next, *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

[96] **Heaps, H.S.** (1978). *Information retrieval: Computational and theoretical aspects*, Academic Press, Inc.

[97] **Gillick, D., Brunk, C., Vinyals, O. and Subramanya, A.** (2015). Multilingual language processing from bytes, *arXiv preprint arXiv:1512.00103*.

[98] **Plank, B., Søgaard, A. and Goldberg, Y.** (2016). Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss, *arXiv preprint arXiv:1604.05529*.

[99] **Ma, X. and Hovy, E.** (2016). End-to-end sequence labeling via bi-directional lstm-cnns-crf, *arXiv preprint arXiv:1603.01354*.

[100] **Sennrich, R., Haddow, B. and Birch, A.** (2015). Neural machine translation of rare words with subword units, *arXiv preprint arXiv:1508.07909*.

[101] **Wu, Y.**, **Schuster, M.**, **Chen, Z.**, **Le, Q.V.**, **Norouzi, M.**, **Macherey, W.**, **Krikun, M.**, **Cao, Y.**, **Gao, Q.**, **Macherey, K.** *et al.* (2016). Google's neural machine translation system: Bridging the gap between human and machine translation, *arXiv preprint arXiv:1609.08144*.

[102] **Ling, W.**, **Trancoso, I.**, **Dyer, C. and Black, A.W.** (2015). Character-based neural machine translation, *arXiv preprint arXiv:1511.04586*.

[103] **Luong, T.**, **Socher, R. and Manning, C.D.** (2013). Better word representations with recursive neural networks for morphology., *CoNLL*, pp.104–113.

[104] **Botha, J. and Blunsom, P.** (2014). Compositional morphology for word representations and language modelling, *International Conference on Machine Learning*, pp.1899–1907.

[105] **Cotterell, R. and Schütze, H.** (2015). Morphological Word-Embeddings., *HLT-NAACL*, pp.1287–1292.

[106] **Ling, W.**, **Luís, T.**, **Marujo, L.**, **Astudillo, R.F.**, **Amir, S.**, **Dyer, C.**, **Black, A.W. and Trancoso, I.** (2015). Finding function in form: Compositional character models for open vocabulary word representation, *arXiv preprint arXiv:1508.02096*.

[107] **Kim, Y.**, **Jernite, Y.**, **Sontag, D. and Rush, A.M.** (2016). Character-Aware Neural Language Models., *AAAI*, pp.2741–2749.

[108] **Wieting, J.**, **Bansal, M.**, **Gimpel, K. and Livescu, K.** (2016). CHARAGRAM: Embedding Words and Sentences via Character n-grams, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp.1504–1515.

[109] **Bojanowski, P.**, **Grave, E.**, **Joulin, A. and Mikolov, T.** (2016). Enriching word vectors with subword information, *arXiv preprint arXiv:1607.04606*.

[110] **Santos, C.D. and Zadrozny, B.** (2014). Learning character-level representations for part-of-speech tagging, *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pp.1818–1826.

[111] **Qiu, S.**, **Cui, Q.**, **Bian, J.**, **Gao, B. and Liu, T.Y.** (2014). Co-learning of Word Representations and Morpheme Representations., *COLING*, pp.141–150.

[112] **Vania, C. and Lopez, A.** (2017). From Characters to Words to in Between: Do We Capture Morphology?, *arXiv preprint arXiv:1704.08352*.

[113] **Gers, F.A.**, **Schmidhuber, J.A. and Cummins, F.A.** (2000). Learning to Forget: Continual Prediction with LSTM, *Neural Comput.*, *12*(10), 2451–2471, http://dx.doi.org/10.1162/089976600300015015.

[114] **Virpioja, S.**, **Smit, P.**, **Grönroos, S.A.**, **Kurimo, M.** *et al.* (2013). Morfessor 2.0: Python implementation and extensions for Morfessor Baseline.

[115] **Alpaydin, E.** (2010). *Introduction to Machine Learning*, The MIT Press, 2nd edition.

APPENDICES

APPENDIX A.1 : Thematic Roles
APPENDIX A.2 : Semantic Roles in Turkish PropBank
APPENDIX A.3 : Definition of Features
APPENDIX A.4 : Label Error Analysis of Finnish
APPENDIX A.5 : Label Error Analysis of German
APPENDIX A.6 : Label Error Analysis of Spanish
APPENDIX A.7 : Label Error Analysis of Catalan
APPENDIX A.8 : Label Error Analysis of Czech
APPENDIX A.9 : Hyperparameters for single unit experiments

APPENDIX A.1

Thematic role	Explanation
Agent	Human or an animate subject that controls or initiates the action.
Patient	Participants that undergoe a state of change.
Theme	Participants in a location or experience a change of location
Beneficiary	Entity that benefits negatively or positively from the action.
Location	Place or path
Destination	End point or direction towards which the motion is directed.
Source	Start point of the motion.
Experiencer	Usually used for subjects of verbs of perception or psychology.
Stimulus	Objects that cause some response from Experiencer.
Instrument	Objects that come in contact with an object and cause a change.
Recipient	Animate or organization target of transfer.
Time	Time.
Topic	Theme of communication verbs.

Table A.1 : Thematic Roles

APPENDIX A.2

SR	Exp	Question shown to Crowdtaskers
A	External Causer	Ettirgen eylemlerdeki yaptıran, ettiren vb
ADV	Adverbial	Tüm cümleyi etkileyen, diğer tanımlara uymayan zarflar (Mutlaka, muhtemelen vb...)
LOC	Location	Nerede (mahallede, konuşmasında, hayalinde)
CAU	Cause	Nedeni ya da kaynağı (yüzünden, onun için, dağdan vb)
LVB	Light Verb	Yardımcı fiil elemanı (mezun olmak'taki mezun, hayal etmek'teki hayal vb...)
COM	Commitative	Kiminle (Kardeşimle, NATOyla, onlarla)
MNR	Manner	Nasıl (hızlıca, güzel, yavaş, yapıp, koşup vb)
DIR	Direction	İzlediği yol (patikadan)
NEG	Negation	Olumsuzluk anlamı ekleyici (Hiçbir zaman, asla, değil, yok, hiç)
DIS	Discourse Connectives	Cümle başındaki Bağlaç (Ayrıca, Fakat, Buna rağmen vb.) ya da Seslenme (Allahım, duy sesimi)
TMP	Time	Ne Zaman (Eylül, Pazartesi), Ne Sıklıkla (her zaman, bazen), Kaçıncı (ilk, son) ya da Ne kadarlığına (bir aylığına)
EXT	Extent	Miktarı (yüzde elli), (az, çok, biraz), (benden fazla) vb.
TWO	Verb Reduplication	Fiil ikilemesi (yapıp yapıp, bakıla bakıla, ister istemez, olursan ol, vb...)
GOL	Goal	Amacı, bitiş noktası (Eve, odaya vb.) ya da faydalanan (annem için, arkadaşıma vb)
INS	Instrument	Ne ile (uçakla, gözleriyle, çekiçle vb...)

Table A.2 : Adjunct Semantic Roles in Turkish PropBank. SR: Semantic Role, Exp: Explanation

APPENDIX A.3

Name	Type	Definition
PredPOS	Syntactic	Predicate POS tag
PredPPOS	Syntactic	Predicate fine-grained POS tag
PredLemma	Lexical	Predicate lemma
PredDeprel	Syntactic	Dependency relation to the predicate
PredFeats	Morphological	Morphological features of the predicate
PredValency	Morphological	Valency of the predicate
PredParentPOS	Syntactic	POS tag of the predicate's head
PredParentFeats	Morphological	Morphological features of the predicate's head
PredParentLemma	Lexical	Lemma of the predicate's head
PredLemmaSense	Semantic	Predicted lemma sense
DepSubCat	Syntactic	Dependency subcategory
ChildDepSet	Syntactic	Set of dependency labels of the children
ChildLemmaSet	Lexical	Set of lemmas of the children
ChildPOSSet	Syntactic	Set of POS tags of the children
ChildCaseMarkerSet	Morphological	Set of case markers of the children
DeprelPath	Syntactic	Dependency path from word to the predeicate
POSPath	Syntactic	POS tag path from word to the predeicate
Distance	Positional	Distance to the predicate
LeftPOS	Syntactic	POS tag of the left sister
RightPOS	Syntactic	POS tag of the right sister
ArgPOS	Syntactic	POS tag of the argument
ArgPPOS	Syntactic	PPOS tag of the argument
ArgDeprel	Syntactic	Dependency relation of the argument
ArgLemma	Lexical	Lemma of the argument
ArgCaseMark	Morphological	Case mark of the argument
ArgMWE	Syntactic	MWE flag for argument's dependency
ArgFirstPosition	Positional	First position flag for argument

Table A.3 : Types and Definitions of Features

APPENDIX A.4

Label	n	char			char3			char5			oracle		
Arg1	1458	0,73	0,71	0,72	0,73	0,71	0,72	0,67	0,64	0,65	0,82	0,78	0,80
Arg2	637	0,64	0,56	0,59	0,60	0,61	0,61	0,54	0,46	0,49	0,63	0,66	0,64
Arg0	554	0,83	0,69	0,75	0,81	0,72	0,77	0,76	0,55	0,64	0,82	0,83	0,83
ArgMtmp	313	0,74	0,61	0,67	0,73	0,64	**0,68**	0,72	0,57	0,64	0,64	0,72	0,67
ArgMmod	174	0,91	0,94	0,92	0,90	0,95	0,92	0,87	0,81	0,84	0,96	1	0,98
ArgMadv	162	0,72	0,31	0,44	0,59	0,37	0,46	0,55	0,37	0,44	0,63	0,40	0,48
ArgMmnr	160	0,54	0,42	0,48	0,56	0,42	0,48	0,42	0,36	0,39	0,52	0,59	0,55
ArgMloc	115	0,55	0,51	0,53	0,47	0,62	0,53	0,44	0,58	0,50	0,57	0,69	0,62
ArgMdis	102	0,69	0,54	0,60	0,69	0,62	0,65	0,71	0,43	0,54	0,69	0,64	0,66
ArgMneg	88	0,87	0,91	**0,89**	0,86	0,91	0,88	0,73	0,68	0,71	0,83	0,93	0,88
Arg3	66	1	0,02	0,03	0,41	0,11	**0,17**	0,20	0,14	0,16	0,44	0,06	0,11
ArgMdir	51	0	0	0	0,27	0,12	0,16	0,21	0,14	**0,17**	0,18	0,06	0,09
ArgMext	44	0,62	0,34	0,44	0,82	0,41	0,55	0,59	0,36	0,45	0,72	0,52	0,61
Arg4	37	0,54	0,19	0,28	0,62	0,14	0,22	0,39	0,24	**0,30**	0,36	0,14	0,20
ArgMcau	36	0,68	0,36	0,47	0,82	0,39	**0,53**	0,58	0,19	0,29	0,50	0,44	0,47
ArgMprt	32	0,83	0,59	0,69	0,71	0,75	0,73	0,79	0,72	**0,75**	0,73	0,75	0,74
ArgMpnc	29	0,36	0,14	0,20	0,32	0,21	0,25	0,08	0,03	0,05	0,47	0,24	0,32
ArgMcsq	16	0,88	0,44	**0,58**	0,88	0,44	**0,58**	0,83	0,31	0,45	0,55	0,38	0,44
ArgMrec	14	0,75	0,43	0,55	0,60	0,43	0,50	0,80	0,57	**0,67**	0,55	0,43	0,48
ArgMprd	10	1	0	0	0	0	0	0	0	0	0	0	0

Table A.4 : **Finnish:** Label errors made by each unit (bi-LSTM) composition

APPENDIX A.5

Label	n	char			char3			char5			oracle		
A0	418	0,75	0,73	0,74	0,75	0,71	0,73	0,71	0,66	0,68	0,79	0,78	0,79
A1	410	0,61	0,63	0,62	0,67	0,68	0,68	0,56	0,59	0,58	0,67	0,72	0,69
A2	138	0,37	0,41	0,39	0,47	0,41	**0,44**	0,29	0,22	0,25	0,43	0,30	0,35
A3	75	0,41	0,35	0,37	0,48	0,43	**0,45**	0,46	0,35	0,39	0,46	0,36	0,40
A4	21	0,20	0,14	**0,17**	0,25	0,10	0,14	1	0	0	1	0	0
A8	4	1	0	0	1	0	0	1	0	0	1	0	0
A5	4	0,33	0,25	**0,29**	1	0	0	1	0	0	1	0	0
A7	3	1	0	0	0	0	0	1	0	0	1	0	0

Table A.5 : **German:** Label errors made by each unit (bi-LSTM) composition

APPENDIX A.6

Label	n	char			char3			char5			oracle		
arg1-pat	2355	0,80	0,83	0,81	0,84	0,82	0,83	0,73	0,61	0,67	0,82	0,88	0,85
arg0-agt	2268	0,81	0,83	0,82	0,85	0,82	0,84	0,68	0,62	0,65	0,82	0,89	0,85
arg1-tem	1717	0,80	0,74	0,77	0,80	0,78	0,79	0,67	0,57	0,62	0,85	0,81	0,83
argM-tmp	1045	0,81	0,73	0,77	0,83	0,75	0,79	0,78	0,58	0,67	0,80	0,80	0,80
arg2-atr	940	0,87	0,83	0,85	0,86	0,85	**0,86**	0,79	0,71	0,75	0,83	0,84	0,84
argM-adv	938	0,54	0,50	0,52	0,57	0,48	0,52	0,53	0,34	0,41	0,54	0,51	0,52
argM-loc	592	0,60	0,71	0,65	0,63	0,72	**0,67**	0,49	0,36	0,42	0,63	0,70	0,66
arg2-null	258	0,61	0,53	0,57	0,66	0,60	**0,63**	0,65	0,47	0,54	0,61	0,62	0,62
arg2-ben	230	0,67	0,58	0,62	0,64	0,72	0,67	0,49	0,16	0,24	0,73	0,74	0,74
argM-mnr	193	0,62	0,28	0,39	0,60	0,28	**0,39**	0,65	0,22	0,33	0,57	0,26	0,36
arg1-null	183	0,53	0,51	0,52	0,68	0,57	0,62	0,71	0,43	0,54	0,63	0,67	0,65
argM-cau	174	0,48	0,45	0,46	0,52	0,51	0,51	0,47	0,29	0,36	0,47	0,59	0,53
arg0-cau	167	0,68	0,37	0,48	0,68	0,50	0,57	0,80	0,19	0,31	0,69	0,64	0,67
arg2-loc	165	0,47	0,28	**0,35**	0,50	0,27	**0,35**	0,40	0,30	0,34	0,37	0,32	0,34
argM-fin	158	0,68	0,61	0,64	0,71	0,59	**0,65**	0,69	0,59	0,64	0,62	0,69	0,65
argM-atr	87	0,55	0,37	0,44	0,58	0,34	0,43	0,43	0,17	0,25	0,64	0,48	0,55
arg4-des	62	0,56	0,73	**0,63**	0,52	0,77	0,62	0,49	0,34	0,40	0,51	0,73	0,60
argL-null	46	0,44	0,15	0,23	0,50	0,28	**0,36**	0,32	0,15	0,21	0,45	0,28	0,35
arg2-ext	33	0,65	0,45	**0,54**	0,55	0,36	0,44	0,60	0,09	0,16	0,50	0,21	0,30
arg2-efi	24	1	0,29	0,45	0,93	0,54	**0,68**	0,79	0,46	0,58	0,78	0,58	0,67
arg3-ori	21	0,32	0,38	0,35	0,36	0,38	0,37	0,33	0,10	0,15	0,35	0,43	0,38
arg4-efi	20	0,60	0,15	0,24	0,44	0,20	**0,28**	1	0	0	0,30	0,15	0,20
argM-ext	19	0,50	0,11	**0,17**	0	0	0	1	0	0	1	0	0
arg3-ben	14	0,50	0,07	0,12	1	0,14	0,25	1	0	0	0,80	0,57	0,67
arg3-ein	9	0,50	0,11	0,18	0,40	0,22	**0,29**	1	0	0	0,25	0,11	0,15
arg2-exp	6	0,33	0,17	0,22	0	0	0	0	0	0	0,40	0,67	0,50
arg1-ext	5	1	0	0	1	0	0	1	0	0	0,67	0,40	0,50
arg3-fin	4	1	0,75	**0,86**	0,60	0,75	0,67	1	0,50	0,67	0,67	1	0,80
arg2-ins	2	1	0	0	1	0	0	1	0	0	0	0	0
arg0-src	2	1	0	0	1	0	0	1	0	0	1	0	0
argM-ins	1	1	0	0	1	0	0	1	0	0	1	0	0
arg1-loc	1	1	0	0	1	0	0	1	0	0	1	0	0
arg0-exp	1	1	0	0	1	0	0	1	0	0	1	0	0

Table A.6 : **Spanish:** Label errors made by each unit (bi-LSTM) composition

APPENDIX A.7

Label	n	char			char3			char5			oracle		
arg1-pat	2444	0,84	0,81	0,83	0,83	0,80	0,82	0,77	0,66	0,71	0,85	0,90	0,87
arg0-agt	2012	0,79	0,81	0,80	0,80	0,78	0,79	0,72	0,58	0,64	0,84	0,89	0,86
arg1-tem	1608	0,80	0,77	0,79	0,80	0,76	0,78	0,72	0,60	0,65	0,85	0,79	0,82
argM-tmp	949	0,82	0,78	**0,80**	0,82	0,77	0,79	0,75	0,59	0,66	0,76	0,74	0,75
arg2-atr	867	0,87	0,81	0,84	0,86	0,81	0,83	0,78	0,70	0,74	0,85	0,82	0,84
argM-loc	686	0,53	0,69	0,60	0,55	0,69	0,61	0,52	0,33	0,40	0,56	0,72	0,63
argM-adv	686	0,52	0,43	**0,47**	0,52	0,38	0,44	0,53	0,30	0,38	0,51	0,42	0,46
arg2-null	405	0,67	0,60	0,63	0,73	0,63	**0,68**	0,66	0,59	0,62	0,66	0,67	0,66
arg1-null	212	0,64	0,49	0,56	0,71	0,54	0,61	0,72	0,50	0,59	0,78	0,58	0,67
arg2-ben	210	0,68	0,51	0,58	0,71	0,49	0,58	0,61	0,27	0,38	0,80	0,46	0,59
argM-cau	191	0,55	0,49	0,52	0,55	0,45	0,49	0,55	0,35	0,43	0,65	0,43	0,52
argM-mnr	173	0,69	0,43	0,53	0,69	0,43	**0,53**	0,78	0,36	0,49	0,77	0,39	0,52
argM-fin	173	0,60	0,58	0,59	0,60	0,57	0,59	0,61	0,43	0,50	0,64	0,56	0,60
arg2-loc	155	0,59	0,17	0,26	0,56	0,21	0,31	0,37	0,20	0,26	0,40	0,32	0,35
arg0-cau	105	0,60	0,39	0,47	0,72	0,41	0,52	0,58	0,20	0,30	0,80	0,42	0,55
argM-atr	92	0,54	0,41	0,47	0,70	0,43	**0,54**	0,59	0,33	0,42	0,64	0,40	0,49
arg4-des	53	0,53	0,53	0,53	0,55	0,60	**0,58**	0,53	0,40	0,45	0,49	0,53	0,51
argL-null	48	0,65	0,35	0,46	0,53	0,38	0,44	0,61	0,23	0,33	0,55	0,44	0,49
arg3-ori	35	0,82	0,40	**0,54**	0,60	0,34	0,44	0,67	0,17	0,27	0,65	0,37	0,47
arg2-ext	28	0,82	0,50	**0,62**	0,52	0,43	0,47	0,50	0,18	0,26	0,46	0,21	0,29
arg2-efi	24	0,90	0,79	0,84	0,95	0,79	**0,86**	0,88	0,62	0,73	0,94	0,67	0,78
arg4-efi	18	0,46	0,33	0,39	0,67	0,33	**0,44**	0,75	0,17	0,27	0,40	0,11	0,17
arg3-ben	14	0	0	0	0,50	0,07	**0,12**	1	0	0	0,50	0,07	0,12
arg2-fin	7	1	0,43	0,60	0,60	0,43	0,50	0,83	0,71	0,77	1	0,71	0,83
arg3-ein	6	0,60	0,50	**0,55**	0,67	0,33	0,44	0	0	0	0,67	0,33	0,44
arg1-ext	5	1	0	0	1	0	0	0	0	0	1	0	0
arg3-atr	4	1	0,75	0,86	1	0,75	0,86	1	0,50	0,67	1	0,75	0,86
arg2-exp	3	0,50	0,33	0,40	0,50	0,33	0,40	1	0,33	0,50	0,67	0,67	0,67
arg3-loc	2	1	0	0	1	0	0	1	0	0	1	0	0
argM-ins	1	1	0	0	1	0	0	1	0	0	1	0	0
argM-ext	1	0	0	0	1	0	0	1	0	0	0	0	0
arg1-loc	1	1	0	0	1	0	0	1	0	0	1	0	0

Table A.7 : **Catalan:** Label errors made by each unit (bi-LSTM) composition

APPENDIX A.8

Label	n	char			char3			char5			oracle		
RSTR	11758	0,92	0,88	0,90	0,92	0,88	0,90	0,90	0,82	0,86	0,94	0,90	0,92
PAT	7265	0,73	0,72	0,72	0,75	0,72	0,73	0,65	0,59	0,62	0,80	0,78	0,79
ACT	6449	0,77	0,73	0,75	0,78	0,73	0,76	0,72	0,55	0,63	0,84	0,81	0,83
APP	2428	0,73	0,80	0,77	0,75	0,82	0,78	0,77	0,71	0,74	0,79	0,84	0,82
LOC	1757	0,70	0,72	0,71	0,71	0,73	0,72	0,60	0,51	0,55	0,75	0,77	0,76
TWHEN	1540	0,79	0,73	0,76	0,82	0,72	0,77	0,75	0,66	0,70	0,80	0,75	0,77
MANN	835	0,71	0,59	0,65	0,73	0,57	0,64	0,69	0,55	0,61	0,73	0,62	0,67
EXT	632	0,79	0,69	0,74	0,83	0,70	0,76	0,80	0,59	0,68	0,86	0,71	0,78
ADDR	573	0,56	0,35	0,43	0,56	0,41	0,48	0,42	0,24	0,30	0,61	0,47	0,53
EFF	557	0,75	0,50	0,60	0,78	0,51	0,62	0,67	0,34	0,45	0,76	0,55	0,64
DIR3	499	0,75	0,59	0,66	0,77	0,61	0,68	0,51	0,32	0,39	0,70	0,68	0,69
MAT	457	0,91	0,77	0,84	0,91	0,84	0,87	0,89	0,73	0,80	0,93	0,84	0,88
ID	363	0,56	0,40	0,47	0,61	0,40	0,48	0,62	0,25	0,35	0,69	0,47	0,56
BEN	361	0,63	0,53	**0,57**	0,67	0,47	0,56	0,60	0,37	0,46	0,61	0,51	0,55
DIR1	359	0,67	0,67	0,67	0,68	0,70	0,69	0,50	0,29	0,36	0,72	0,68	0,70
ACMP	324	0,55	0,40	0,46	0,60	0,44	0,51	0,44	0,24	0,31	0,66	0,46	0,54
REG	285	0,69	0,27	0,39	0,71	0,28	**0,40**	0,70	0,21	0,33	0,67	0,26	0,37
MEANS	251	0,46	0,34	0,39	0,60	0,38	0,47	0,46	0,20	0,28	0,57	0,49	0,53
CPHR	224	0,83	0,49	0,61	0,86	0,56	0,68	0,76	0,53	0,63	0,86	0,61	0,71
CAUS	219	0,72	0,46	0,56	0,69	0,47	0,56	0,72	0,30	0,42	0,79	0,44	0,57
COND	206	0,64	0,53	**0,58**	0,65	0,52	**0,58**	0,47	0,18	0,26	0,71	0,46	0,55
CRIT	202	0,85	0,75	0,80	0,88	0,78	0,82	0,88	0,66	0,75	0,90	0,80	0,85
AIM	184	0,55	0,29	0,38	0,43	0,31	0,36	0,51	0,22	0,31	0,45	0,35	0,40
THL	162	0,74	0,61	0,67	0,68	0,63	0,65	0,70	0,52	0,60	0,71	0,63	0,67
COMPL	137	0,60	0,61	0,60	0,59	0,66	0,62	0,59	0,49	0,54	0,65	0,67	0,66
THO	103	0,77	0,71	0,74	0,78	0,71	0,74	0,82	0,68	0,74	0,87	0,74	0,80
DPHR	103	0,80	0,35	0,49	0,81	0,33	0,47	0,71	0,34	0,46	0,79	0,36	0,49
TTILL	97	0,80	0,74	**0,77**	0,76	0,77	**0,77**	0,73	0,41	0,53	0,79	0,69	0,74
ORIG	87	0,45	0,24	0,31	0,51	0,23	0,32	1	0	0	0,53	0,30	0,38
COMPL2	83	0,33	0,14	**0,20**	0,29	0,11	0,16	0,36	0,05	0,09	0,28	0,06	0,10
DIFF	82	0,68	0,57	0,62	0,80	0,63	0,71	0,78	0,22	0,34	0,80	0,68	0,74
TSIN	74	0,78	0,70	0,74	0,81	0,73	**0,77**	0,62	0,22	0,32	0,80	0,72	0,76
AUTH	71	0	0	0	0,44	0,10	0,16	0,33	0,04	0,07	0,52	0,20	0,29
CPR	64	0,68	0,36	0,47	0,71	0,47	**0,57**	0,75	0,28	0,41	0,78	0,44	0,56
CNCS	63	0,62	0,54	**0,58**	0,76	0,41	0,54	0,60	0,05	0,09	0,74	0,32	0,44
TPAR	56	1	0,38	0,55	0,92	0,41	0,57	0,80	0,29	0,42	0,81	0,39	0,53
RESTR	45	0,91	0,71	**0,80**	0,85	0,73	0,79	0,74	0,51	0,61	0,85	0,73	0,79
DIR2	43	0,14	0,05	0,07	0,21	0,07	**0,11**	0,25	0,02	0,04	0,27	0,07	0,11
RESL	34	0,62	0,38	0,47	1	0,21	0,34	1	0,03	0,06	0,75	0,35	0,48
TFHL	29	0,35	0,28	**0,31**	0,36	0,28	**0,31**	0,38	0,21	0,27	0,29	0,21	0,24
TOWH	21	0,40	0,19	0,26	0,43	0,14	0,21	1	0	0	0,70	0,33	0,45
ACT|ADDR	21	1	0	0	1	0	0	1	0	0	1	0	0
SUBS	20	0,50	0,10	0,17	0,50	0,10	0,17	1	0,05	0,10	0,62	0,25	0,36
ACT|PAT	20	1	0	0	1	0	0	1	0	0	1	0	0
ACT|COMPL	20	1	0,05	**0,10**	1	0	0	1	0	0	0	0	0
INTT	15	1	0	0	1	0	0	1	0	0	1	0	0
TFRWH	14	0,20	0,07	0,11	0,45	0,36	**0,40**	0,20	0,07	0,11	0,33	0,21	0,26
CONTRD	14	0,38	0,64	**0,47**	0,35	0,57	0,43	0,33	0,21	0,26	0,29	0,43	0,34

Table A.8 : **Czech:** Label errors made by each unit (bi-LSTM) composition

APPENDIX A.9

Name	Value
k (for cross validation fold)	10
parameter initialization method	uniform
initialization range	$-0,1;+0,1$
optimization method	stochastic gradient descent
gradient clipping (max gradient)	2
dropout	0,50
learning rate	1
decay rate	0,50
patience	3
epochs	50
subword bi-LSTM hidden size	200
subword bi-LSTM layer size	2
char dimension	200
word dimension	200
morpheme dimension	200
SRL bi-LSTM hidden size	128
SRL bi-LSTM layer size	1
use region mark	False
use binary mask	True
batch size	32
maximum sequence length	200

Table A.9 : Hyperparameters for single unit experiments

CURRICULUM VITAE

Name Surname: Gözde Gül ŞAHİN

E-Mail: goezde.guel@gmail.com

EDUCATION:

- **PhD :** 2018, Istanbul Technical University, Faculty of Electrical and Electronics Engineering, Computer Engineering
- **M.Sc.:** 2011, Sabancı University, Faculty of Engineering and Science, Electronics Engineering
- **B.Sc.:** 2009, Istanbul Technical University, Faculty of Electrical and Electronics Engineering, Computer Engineering

PROFESSIONAL EXPERIENCE:

- May 2017 - Nov 2017, Visiting Researcher at Institute for Language, Logic and Computation (ILCC), University of Edinburgh
- May 2016 - May 2017, PT Researcher at Yapı Kredi Bankası for "Relation Extraction from Banking Documents"
- Aug 2016 - May 2017, R&D Engineer at PragmaCraft
- Nov 2014 - May 2016, Project Employee at Turkey State Planning Organization Funded Project, entitled "English to Turkish Statistical Machine Translation System"
- May 2011 - Oct 2013, Software Engineer at Siemens Corporate Technologies and Development Center
- Mar 2009 - Mar 2011, Teaching Assistant at Sabancı University

HONORS, AWARDS and SCHOLARSHIPS:

- Tübitak 2224A Travel Grant, To attend LxMLS 2016 (Lisbon Machine Learning School), June 2016
- Tubitak 2214A Research Abroad Scholarship, To conduct research on Turkish Semantic Role Labeling in University of Edinburgh, November 2016
- LDC (Linguistic Data Consortium) Data Scholarship, Fall 2015

- Tubitak 2211A Domestic PhD Scholarship, 2013 - 2017
- Istanbul Technical University Grant, 2004 - 2009, Given for taking place in the first 1500 in nationwide university entrance exam

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- Gözde Gül Şahin, Eşref Adalı, Annotation of semantic roles for the Turkish Proposition Bank., *Journal of Language Resources and Evaluation*, pp.1-34., May 2017
- Gözde Gül Şahin, Eşref Adalı, Verb Sense Annotation For Turkish PropBank via Crowdsourcing, *7th International Conference on Intelligent Text Processing and Computational Linguistics (CICLING 2016)*, Konya, Turkey, April
- Gözde Gül İşgüder, Eşref Adalı, Using Morphosemantic Information in Construction of a Pilot Lexical Semantic Resource for Turkish, *In Proceedings of Workshop on Lexical and Grammatical Resources for Language Processing (COLING 2014)*, Dublin, Ireland

OTHER PUBLICATIONS, PRESENTATIONS AND PATENTS:

- Gözde Gül Şahin, Mark Steedman, Cross-Predicate Sense Disambiguation, *Woman in Machine Learning Workshop (WiML), NIPS 2017*, Long Beach, USA, December

- Gözde Gül Şahin, Erdem Emekligil, Seçil Arslan, Onur Ağın, Gülşen Eryiğit, Relation extraction via one-shot dependency parsing on inter-sentential, higher-order and nested relations, *Journal of Turkish Electrical Engineering & Computer Sciences*, June 2017

- Gözde Gül Şahin, Erdem Emekligil, Seçil Arslan, Onur Ağın, and Gülşen Eryiğit. 2017. Conversion of number expressions within noisy text into numerical representation. *In 25. Sinyal İşleme ve İletişim Uygulamaları Kurultayı SIU*, Antalya, Turkey, June

- Gözde Gül Şahin, Framing of Verbs for Turkish PropBank, *International Conference on Turkic Computational Linguistics at CICLING 2016*, pages 12–17, Konya, Turkey, April

- Gözde Gül Şahin, Harun Reşit Zafer, Eşref Adalı, Polarity Detection of Turkish Comments on Technology Companies, *In Proceedings of International Conference on Asian Language Processing 2014 (IALP 2014)*, Kuching, Malaysia, October

- Gözde Gül Şahin, Eşref Adalı, A Pilot Study on Automatic Inference Rule Discovery from Turkish Text, *In Proceedings of 8th International Conference on Application of Information and Communication Technologies (AICT 2014)*, Astana, Kazakhstan, October

- Gözde Gül İşgüder, Gozde Unal, Martin Groher, Nassir Navab, Ali Kemal Kalkan, Muzaffer Degertekin, Holger Hetterich and Johannes Rieber, Manifold Learning for Image-Based Gating of Intravascular Ultrasound (IVUS) Pullback Sequences, *Medical Imaging and Augmented Reality, (MIAR 2010)*, Bejing, China, September

- Serhan Gurmeric, Gözde Gül İşgüder, Gozde Unal, Stephane Carlier, A New 3-D automated computational method to evaluate in-stent neointimal hyperplasia in in-vivo intravascular optical coherence tomography pullbacks, *12th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI), 2009*, London, UK, September