ISTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

Turkish Morphological Disambiguation using Multiple Conditional Random fields

M.Sc. THESIS

Razieh Ehsani

Department of Computer Engineering

Computer Engineering Programme

Thesis Supervisor: Prof. Dr. Eşref ADALI

JUNE 2012

ISTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

Turkish Morphological Disambiguation using Multiple Conditional Random fields

M.Sc. THESIS

Razieh Ehsani
(504091523)

Department of Computer Engineering

Computer Engineering Programme

Thesis Supervisor: Prof. Dr. Eşref ADALI

JUNE 2012

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

Birçok Koşullu Rassal Alan Kullanarak Türkçe için Biçimbilimsel Belirsizlik Giderme

YÜKSEK LİSANS TEZİ

Razieh Ehsani
(504091523)

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Eşref ADALI

Haziran 2012

Razieh Ehsani, a M.Sc. student of ITU Institute of Science and Technology 504091523 successfully defended the thesis entitled **"Turkish Morphological Disambiguation using Multiple Conditional Random fields"**, 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	.....
Co-advisor :	**Yrd. Doç. Gülşen Cebiroğlu Eryiğit**	.....
Jury Members :	**Prof.Dr.Eşref Adalı**	.....
	Assoc.Prof.	.....
	Assoc.Prof	.....

Date of Submission : **30 JUNE 2012**
Date of Defense : **2 August 2012**

FOREWORD

...

June 2012

Razieh Ehsani
Computer Engineer

TABLE OF CONTENTS

Page

ABBREVIATIONS

NLP	**:** Natural Language Processing
CRF	**:** Conditional Random Fields
HMM	**:** Hidden Markov Model
MEMM	**:** Maximum Entropy Markov Model
MD	**:** Morphological Disambiguation
POS	**:** Part of Speech

LIST OF TABLES

LIST OF FIGURES

Page

Turkish Morphological Disambiguation using Multiple Conditional Random fields

SUMMARY

We use a statistical approach to tackle the morphological disambiguation problem. The Conditional random fields (CRFs) are a class of statistical modeling methods widely used in several NLP tasks. Compared with the other statistical approaches such as Hidden Markov Models and Maximum Entropy Markov Models, we use CRFs because it is more compatible with the nature of the morphological disambiguation problem. Also, CRFs are robust to over-fitting problem, since the number of parameters of the model is relatively less.

CRFs can solve Label Bias problem because the normalization is performed at the sentence level. Furthermore, the likelihood function is convex, which means the global optimum can always be found using gradient based methods. Consequently, CRF is a successful method for sequence classification. Also CRFs can explicitly specify desired conditional dependencies We define the linguistic features for our modeling. These features will defined as edge features and node features on the rest of paper. We use minimum Redundancy Maximum Relevance (mRMR) algorithm for choosing relevance features.

Birçok Koşullu Rassal Alan Kullanarak
Türkçe için Biçimbilimsel Belirsizlik Giderme

ÖZET

Hesaplamalı dil bilim bir dilin yapısal ve ya istatistiksel özelliklerini incelemek ve dile ait verileri işleyerek,belli başlı sorunlara çözüm arıyan, disiplinler arası bir bilim dalıdır.Bu disiplinler arasında önde gelenleri bilgisayara bilimler, dil bilimleri,bilişsel bilimler ve felsefe gelmektedir.Hesaplamalı dil bilimlerinde amaç dilin yapısal özelliklerine dayır kuramsal çıkarımlar yapmak olabilmekle birlikte dili modellemek ve işlemek suretile uygulamada bazı faydalar elde etmekte olabilir.İlk çalışmalar 1950 yıllarda makine tercüme alanında başlamıştır.

Doğal dil işleme de hesaplamalı dil bilimlerinin önemli konularından bir tanesidir,burada amaç dili pratik bir amaca hizmet etmek için modellemektir,kuramsal hesaplamalı dil bilimsel çalışmalardan faklı olarak,doğal dil işemede,dilinin modelenmesindeki karmaşıklık,hizmet edecek amaca uygun olarak değişe bilir,dolayısıyla burada amaç dili mümkün olduğunca iyi modellemek değil istenen amacı mümkün olduğunca başarılı bir şekilde gerçekliştirmektir.Makine çevirisi,biçimbilimsel inceleme,biçimbilimsel belirsizlik giderme,anlamsal belirsizlik giderme,bilgi çıkarımı gibi konular doğal dil işlemenin önemli konuları arasındadır.Genelde iki temel yaklaşım olduğu gözlene bilir. Bunlardan ilki dilin belirli önemli yapısal özelliklerini öne çıkararak,elle belirlenen ve ya otomatik çıkarılan kurallar yoluyla istenen amacı gerçekleştirilir.Diğer bir yaklaşım ise dili çeşitli gelişmiş istatistik ve makine öğrenmesi yöntemleri ile modellemektir. Bizim çalışmamız bu ikinci yaklaşımı benimsemektedir.

Özellikle karmaşık biçimbilimsel özellikler gösteren dillerde (Türkçe,Fince,)biçimbilimsel analiz ve belirsizlik giderme konuları önemlidir.Biçimbilimsel belirsizlik giderici Türkçe'de diğer doğal dil işleme konularında bir önişleme olarak ele alınmaktadır. Türkçe'de biçimbilimsel belirsizlik Türkçenin zengin biçimbilimsel özelliğinden kaynaklanıyordur,Türkçe bir kelime teorik olarak sonsuz ek alabilmektedir,aldığı her ek ile kelimenin biçimbilimsel özelliği değişebiliyordur,bu zenginlikle birlikte cümle içindeki aldığı pozisiyonda bu belirsizliğe daha çok neden oluyordur. Bazı Türkçe kelimelerin 20 üzerinde biçimbilimsel analize sahip olduklarını görebiliyoruz. Biçimbilimsel analiz Türkçe'de 116 etiketten oluşmaktadır.Her kelime cümlede konumunabakmaksızın bu etiketlerden oluşan bir katarı biçimbilimsel analiz olarak alıyordur.Bu etiketlerden 12'si Part of Speech olarak,kelimenin sıfat,isim, fiil ve ... olmasını belirtiyor. Biçimbilimsel belirsizlik giderici, kelimenin biçimbilimsel analizlerinden cümlede aldığı konumuna göre doğru olanı seçme yöntemidir.Ana etiket belirsizliği giderme ise kelimenin cümlede aldığı konuma göre alacağı ana etiketler kümesinden alabileceği etiketi belirlemektir.Bu sorun İngilizce gib dillerde çok karmaşık bir problem değildir ama

Türkçede ise zor bir soruna dönüşüyordur. Bizim çalışma Türkçenin hem ana etiket belirsizliği hem aynı zamanda biçimbilimsel belirsizliği gidermektir.Bu sorunu daha önce yapılan çalışmalardan farklı olarak istatistiksel makine öğrenmesi yöntemi ile ele almaktayız.Son zamanların doğal dil işleme çalışmalarında yer alan koşullu rassal alanlar yöntemini bu çalışmada kullanılmıştır. Koşullu rassal alanlar,bir koşullu olasılık dağılımıdır.Koşullu rassal alanlar yaklaşımında biçimbilimsel analizleri kelimelere koşullu olarak bir olasılık atamaya çalışılıyor,biçimbilimsel analizlerin arasındaki her hangi bir biçimbilimsel ve ya istatistiksel ilişkileri koşullu rassal alanlar özellik olarak kullanıyor,Ayrıca biçimbilimsel analizlerin ve kelimelerin arasındaki istatistiksel ve biçimbilimsel özellikleri de kullanıyor. Bu özelliklerin ağırlıklarını öğrenme verisinden öğrenmektedir.Bu çalışmanın temel konularından bu özelliklerin tanımı ve yararlı özelliklerin seçilmesidir.Doğru özellikler başarıyı daha yükseltiyor. Koşullu rassal alanlar parametre öğrenmede L-BFGS algoritmasını ve çıkarım kısımında ise viterbi algoritmasını kullanıyor.Bu çalışmada zincir koşullu rassal alanlar kullanılıyor.Zincir koşullu rassal alanlar bir graftaki komşulukları göze almaktadır. Öğrenme ve deneme amaçlı mallet aracını kullandık.Bu araç ayrıca koşullu rassal alanların doğası gereği yavaş ve zaman alıcı bir araçtı, bu çalışmada ayrıca daha başarılı ve daha hızlı sonuca varmak için çeşitli yöntemler geliştirilmiştir. Bu yöntemler, hem ana etiket atama probleminde hem biçimbilimsel belirsizlik giderici de koşullu rassal alanların cümle bazında optimizasyon yapmasını mümkün kılmıştır ve bu sebebten dolayı başarıyıda ayrıca yükseltmiştir.Ana etiket atama probleminde bir tek koşullu rassal alan kullanılırken biçimbilimsel belirsizlik gidericide bir çok koşullu rassal alan kullanılmıştır.Biçimbilimsel belirsizlik gidericide 116 etiketi 9 ayrı kümede toplamıştır. Bu 9 küme,Türkçenin biçimbilimsel özelliğine göre düzellenmiştir.Biçimbilimsel belirsizlik gidericide bu 9 küme için ayrı ayrı koşullu rassal alanlar eğitilmiştir.Bu eğitilmiş koşullu rassal alanları sonra birleştirip ve problemi çözüyoruz. Ana etiket atama probleminde 98.5 civarında bir başarı elde edilmiştir.

1. INTRODUCTION

1.1 Purpose of Thesis

Communication is a crucial part of any social organization and as the technology advances, the benefits are also noticed in this area. The benefits include the ability to communicate further and faster than before and with more people simultaneously. But the advanced technology does not serve only as a more efficient carrier of human communication signals but also as effective processors of these signals. It is the task of the field of Natural Language Processing (NLP) to derive important characteristics of a communication signal and consequently process it.

The existence of different models of communications, such as different languages, creates various challenges. Sometimes, it might not be possible to derive a method that works best for any language, in such problems domain/language dependent studies are due. Turkish language has a very rich morphological structure and as an agglunative language, shows very different characteristics compared to, say, English. One example is the property that the words in Turkish can, theoretically, take infinite suffixes and a suffix may change the semantic or syntactic properties of a given word drastically.

Analysing morphological properties of a given sentence is a crucial task for many subsequent processing, such as parsing and word-sense disambiguation and many other supervised methods in NLP. However, when the analysis is done at a word level, one usually gets multiple possible analyses to choose from. A full analysis of a word contains morphological properties such as Part-of-Speech (POS) tag, tense, plurality, etc. This problem of ambiguity can only be solved using contextual information in terms of the sentence involved or maybe even a larger unit.

The ratio of ambiguous words to all words is 50% in our corpus of Turkish sentences, that means a morphological analyser will fail to unambigously identify the correct analysis using the word features alone. Moreover, some of the words in the corpus can

have up to 23 different analysis. This complex structure of Turkish language in terms of its morphological properties makes the morphological disambiguation problem a highly difficult one.

The morphological disambiguation problem for morphologically rich languages differs significantly from the well known POS tagging problem. It is rather an automatic selection process from multiple legal analysis of a given word than the assignment of a POS tag from a predetermined tag set. The possible morphological analyses of a word (generally produced by a morphological analyzer) in such languages are very complex when compared to morphologically simple ones: They consist of the lemma, the main POS tags and the tags related to the inflectional and derivational affixes. The number of the set of possible morphological analyses may sometimes be infinite for some languages such as Turkish.

In this study, we focus on the determination of the main POS tags (which will be referred as "POS tagging" from now on) and in next step the full disambiguation task. There are few methods for Turkish which directly tackle POS tagging problem. Instead many methods perform a full morphological disambiguation and the POS tags are obtained from the correct parses. In this work, we take a different approach and propose a model which directly tackles the POS tagging problem. While also being useful in its own right, this method is also a first step towards full morphological disambiguation through weighted opinion pooling approach [1]. In the other step we focus on the morphological disambiguation problem.

To give a sense of the problem at hand and the general morphological disambiguation, we have measured the ambiguity corresponding to the POS tagging and Morphological Disambiguation problems. About 27% of the words in our corpus are ambiguous in terms of its POS tag and random guessing has an expected accuracy of 85%, on the other hand the ambiguity in terms of morphological disambiguation is about %50. The proposed approach in this paper improves the accuracy of POS tag to around 98.48%.

Our approach is based on the well known methodology of Conditional Random Fields, which is also applied to other languages with varying success. POS tagging problem was successfully tackled in languages with relatively simpler morphological properties

(such as English) [1, 2, 3, 4]. On the other hand, other languages proved to be more problematic with lower tagging performance, [5, 6, 7] with accuracies ranging from %85 to %95. Smith et. al. [1] discusses the high computational burden of CRFs in both training and inference steps and argues that this is a major obstacle in its practical usage. In this work, we also discuss performance related issues and propose different approaches to lower the computational burden in inference step. The best approach among these approaches the state of the art [8] in performance, while being competitive in computational complexity. We also discuss the problem of feature selection in order to reduce training times and improve generalization capability. We employ the well known mRMR [9] method to this end. These efficiency improvements are important steps toward making CRFs more practical tools in NLP.

2. BACKGROUND ON MORPHOLOGICAL DISAMBIGUATION

In this chapter, we shall introduce the Turkish morphological properties.

2.1 Morphological Properties of Turkish Sentences

Turkish is an agglunative language which has a complex morphological structure. This property of the Turkish language leads to vast amounts of different surface structures found in texts. In a corpus of ten million words, the number of distinct words exceeds four hundred thousand [**?**]. There are several suffixes, which may change the POS tags of the words from noun to verb or verb to adverb, etc. Thus, it is much harder to determine the final POS tag of a word using the root such as in English. Because of this, we can not resort to lexicons of words (roots) as in many studies on English. We must use the morphological analysis of the words to determine the tags. The context dependency of tags of words must also be taken into account.

There are several tags which determine respective properties of the associated words. These tags contain syntactic and semantic information and are called morphosyntactic or morphosemantic respectively. We use the same representation for the tags as [10]. Any words in Turkish can be represented by the chain of these tags. We call these chains of tags for words morphological analyses of these words.

Turkish morphological analysis considers 116 different tags. To better model these tags and circumvent the data sparseness problems, we have partitioned these into 9 disjoint groups, called slots. The slots are determined such that the semantic relation among the tags in a slot is maximum, while it is minimum for tags across slots. Also a word can not accept more than one tag from a single slot. Essentially transforming the problem into a multiple class classification problem. Such a construction of the problem, with this particular slot partitioning, is one of the contributions of the paper. The main properties of the words are expressed in the main POS category and the other slots serve to fill in the details such as plurality, tense, etc. In this paper, we are

concerned with the correct disambiguation of the main POS tags, so we are interested in identifying the value of a single slot. However, the other slots serve as features in our models, which will be discussed in detail in later sections.

Many words in Turkish texts have more than one analysis. Sometimes the number of analyses reach 23. Because of the Turkish language derivative and inflective property, in theory, one word can use an infinite number of suffixes. Due to this, we are faced with immense vocabulary in Turkish. The large vocabulary size causes data sparseness problem. Some of these suffixes change the word meanings. In this case, these changes are expressed with inflectional groups (IGs) that are separated by *ˆDB* sign, where *ˆDB*'s mean derivation boundary (root+IG1+ ˆDB+IG2+ ˆDB+...+ ˆDB+IGn). One Turkish word can have many IGs in its analyzes. These IGs and the related tags can also be represented as tags. The standard morphological tags, also used in this work, are shown in Table 2.1. The example below shows the analyses for the word "alındı" produced by a Turkish two-level morphological analyzer [14].

1. al+VerbˆDB+Verb+Pass+Pos+Past+A3sg (It was taken)
2. al+AdjˆDB+Noun+Zero+A3sg+P2sg+NomˆDB+Verb+Zero+Past+A3sg (It was your red)
3. al+AdjˆDB+Noun+Zero+A3sg+Pnon+GenˆDB+Verb+Zero+Past+A3sg (It was the one of the red)
4. alındı+Noun+A3sg+Pnon+Nom (receipt)
5. alın+Verb+Pos+Past+A3sg (resent)
6. alın+Noun+A3sg+Pnon+NomˆDB+Verb+Zero+Past+A3sg (It was the forhead)

We categorized all tag in morphological analyzes into 9 independent groups. First group is main part of speech group, that determine general morphological properties of words. Each tag determine major POS of word as :

1. **Noun:** Noun
2. **Adj:** Adjective

Slot Groups	Slot Values
Main POS	Adj, Adv, Conj, Det, Dup, Interj, Noun, Num, Postp, Pron, Punc, Verb
Minor POS	Able, Acquire, ActOf, Adamantly, AfterDoingSo, Agt, Almost, As, AsIf, AsLongAs, Become, ByDoingSo, Card, Caus, DemonsP, Dim, Distrib, EverSince, FeelLike, FitFor, FutPart, Hastily, InBetween, Inf, Inf1, Inf2, Inf3, JustLike, Ly, Ness, NotState, Ord, Pass, PastPart, PCAbl, PCAcc, PCDat, PCGen, PCIns, PCNom, Percent, PersP, PresPart, Prop, Quant, QuesP, Range, Ratio,Real, Recip, ReflexP, Rel, Related, Repeat, Since, SinceDoingSo, Start, Stay, Time, When, While, With, Without, Zero
Person Agreements	A1pl, A1sg, A2pl, A2sg, A3pl, A3sg
Possessive Agreements	P1pl, P1sg, P2pl, P2sg, P3pl, P3sg, Pnon
Case Markers	Abl, Acc, Dat, Equ, Gen, Ins, Loc, Nom
Polarity	Neg, Pos
Tense/Mood	Aor, Desr, Fut, Imp, Neces, Opt, Pres, Prog1, Prog2, Cop, Cond, Past, Narr
Compund Tense	Comp_Cond, Comp_Narr, Comp_Past
Cop	Cop

Table 2.1: Morphological Tags

3. **Adv:** Adverb
4. **Cond:** Condition
5. **Det:** Determiner
6. **Dup:** Duplicator
7. **Interj:** Interjection
8. **Verb:** Verb
9. **Postp:** Postpositive
10. **Num:** Number
11. **Pron:** Pronoun
12. **Punc:** Punctuation

Second group is Minor Parts of Speech of word,taht is cantent 65 morphological tags, these tags determine the minor morphological properties such as semantic markers, causative markers, postposition and .... We listed these tag with their discribtion as :

1. **Able:** able to verb
2. **Acquire:** to acquire the noun in the stem
3. **ActOf**
4. **Adamantly**
5. **AfterDoingSo:**
6. **Agt:** someone involved in some way with the stem noun
7. **Almost:** almost verbed but did not
8. **As**
9. **AsIf**
10. **AsLongAs**
11. **Become:** to become like the noun or adj in the stem
12. **ByDoingSo**
13. **Card:** Cardinal
14. **Caus:** Causative
15. **DemonsP:** Demonstrative Pronoun
16. **Dim:** Diminutive
17. **Distrib:** Distribution
18. **EverSince:** have been verbing ever since
19. **FeelLike**

20. **FitFor**
21. **FutPart:** Future Participle
22. **Hastily:** verb hastily
23. **InBetween**
24. **Inf:** Infinitive
25. **Inf1:** Infinitive
26. **Inf2:** Infinitive
27. **Inf3:** Infinitive
28. **JustLike**
29. **Ly:** corresponds to English slow,slowly
30. **Ness:** as in Red vs Redness
31. **NotState**
32. **Ord:** Ordinal
33. **Pass:** Passive
34. **PastPart:** Past Participle
35. **PCAbl**
36. **PCAcc**
37. **PCDat**
38. **PCGen**
39. **PCIns**
40. **PCNom**
41. **Percent:** Percentage

42. **PersP:** Personal Pronoun

43. **PresPart:** Present Participle

44. **Prop:** Porper Noun

45. **Quant:** Quantifying

46. **Ques:** Question

47. **Range:** Range

48. **Ratio:** Ratio

49. **Real:** Real

50. **Recip:** Peciprocal

51. **ReflexP:** Reflexsiv Pronoun

52. **Rel**

53. **Related**

54. **Repeat** verb repeatedly

55. **Since**

56. **SinceDoingSo**

57. **Start:** start verbing immediately

58. **Stay:** stayed/frozen while verbing

59. **Time:** Time

60. **When:**

61. **While:**

62. **With:**

63. **Without:**

64. **WithoutHavingDoneSo:**

65. **Zero:**

Third group is Number/Person Agreement that indicated by the following:

1. **A1sg:** 1. singular
2. **A2sg:** 2. singular
3. **A3sg:** 3. singular
4. **A1pl:** 1. plural
5. **A2pl:** 2. plural
6. **A3pl:** 3. plural

Fourth group is Possessive Agreement is indicted by the following:

1. **P1sg:** 1. singular
2. **P2sg:** 2. singular
3. **P3sg:** 3. singular
4. **P1pl:** 1. plural
5. **P2pl:** 2. plural
6. **P3pl:** 3. plural
7. **Pnon:** Pronoun (no overt agreement)

Fifth group is Case Marker group is indicated by the following:

1. **Nom:** Nominative
2. **Acc:** Accusative/Objective
3. **Dat:** Dative (to ...)

4. **Abl:** Ablative (from ...)
5. **Loc:** Locative (on/at/in ...)
6. **Gen:** Genitive (of ....)
7. **Ins:** Instrumental (with ...)
8. **Equ:** Equative (by (object) in passive sentences)

Sixth group is verb markers as Polarity:

1. **Pos:** Positive
2. **Neg:** Negative

Seventh group is Tense and Aspect and Mood group:

1. **Past:** Past tense
2. **Narr:** Narrative past tense
3. **Fut:** Future tense
4. **Aor:** Aorist, may indicate, habitual, present, future you name it
5. **Pres:** Present tense
6. **Desr:** Desire/wish
7. **Cond:** Conditional
8. **Neces:** Necessitative, must
9. **Opt:** Optative, let me/him/her verb
10. **Imp:** Imperative
11. **Prog1:** Present continuous, process
12. **Prog2:** Present continuous, state

Verbs may have 1 or 2 such markers, we model the morphological tags given the input sentence conditionaly independent we seprate +Cop in another group also for +Past, +Narr, +Cond we indicated aditionaly group if one verb get these tags togather.

For example we show analyzes and IGs of the word "karın" in below:

1. kar+Noun+A3sg+Pnon+Gen (of the snow)
2. kar+Noun+A3sg+P2sg+Nom (your snow)
3. kar+Verb+Pos+Imp+A2pl (mix)
4. karı+Noun+A3sg+P2sg+Nom (your wife)
5. karın+Verb+Pos+Imp+A2sg (be mixed)
6. karın+Noun+A3sg+Pnon+Nom (stomach)

2.1.0.1 Evaluate POS tagging perforance of Haşim Sak's work

In Sak's POS tagger by using metu_sabanci treebank test set we have these statistics as below:

All word count	45999
Correct word count	39048
Incorrect word count	6951
accuarcy	84.89

Table 2.2: Statistical Analyzes

the case with the greatest number of errors :

POS tag	**Count**	**Incorrect**	**Accuarcy**
P3sg	11361	868	7.64
Adj	9153	800	8.7
Adv	6126	566	9.2393
P3Pl	938	417	42.4211
Pron	2267	308	13.586
Num	707	226	31.98
Persp	1157	261	22.55
P2sg	329	168	51.06

Table 2.3: Error Analyzes

POS tag	Incorrect POS	Mismatch count
Adj	Noun	501
Adj	Verb	139
Adj	Det	99
Adj	Adv	42

Table 2.4: Error Analyzes for Adj

Table 2.3 Example for P3sg's error : [8]. (pantolonu : P3sg|A3sg) dizlerine dek ıslak

Table 2.4 Example for Adj's error : [8]

Ercan Tezer , (içl Adj|Noun) pazarda bu yıl güzelliğini bile fark(edemez| Adj|Verb) hale gelmiştim Gönlüm sizin bu kadar (çok|Adj|Det) acı çekmenize razı değil çocuklar ihtiyaçları göz önüne alındığında (çok|Adj|Adv) hararetli tüketicilerdir

POS tag	Incorrect POS	Mismatch count
Adv	Adj	305
Adv	Noun	97
Adv	Det	68
Adv	Postp	57

Table 2.5: Error Analyzes for Adv

Table 2.5 Example for Adv's error : [8]

Kaç gündür bu (böyle|Adv|Adj) . Kumral saçları (hafifçe|Adv|Adj) karışmıştı . Bence yeterince değil , hiç (araştırmadan|Adv|Noun) haber yapılmış kaçtığını anlar , (bir|Adv|Det) daha herkes gibi içtenlikle her şeyden (önce Adv|Postp) herhangi bir keyfi iradeden bağımsız

POS tag	Incorrect POS	Mismatch count
Pron	Det	152
Pron	Adj	72
Pron	Noun	57
Pron	Adv	27

Table 2.6: Error Analyzes for Pron

Table 2.6 Example for Pron's error : [8] Kaç gündür (bu|Pron|Det) böyle tabanı ne derse (o|Pron|Det) olacak Oğlum , (Ne| Pron|Adj) işe yaradığını , sordu . Biliyorum işte ! (O|Pron|Noun) benim makinem Bilmeyecek (ne| Pron|Adv) var ? Table 2.7

POS tag	Incorrect POS	Mismatch count
Num	Noun	113
Num	Det	111

Table 2.7: Error Analyzes for Num

Example for Num's error : [8] ihracat bedellerinin (yüzseksen|Num|Noun) gün içinde yurda kutularından (bir|Num|Det) iki tane

3. FEATURE SELECTION

3.1 Feature Selection

One method to improve the performance of a machine learning method is to select a subset of informative features [15]. A good feature selection method can improve variance of the estimates without introducing a significant bias. The minimum Redundancy Maximum Relevance (mRMR [9]) method relies on the intuitive criteria for feature selection which states that the best feature set should give as much information regarding the class variable as possible while at the same time minimize inter-variable dependency as much as possible (avoiding redundancy). The two concepts, relevancy and redundancy, can be naturally expressed using information theoretic concept of mutual information. However, real data observed in various problems are usually too sparse to correctly estimate the joint probability distribution and consequently the full mutual information function. The solution proposed in [9], employs two different measures for redundancy (*Red*) and relevance (*Rel*):

$$Red = 1/|S|^2 \sum_{F_i,F_j \in S} MI(F_i,F_j) \tag{3.1}$$

$$Rel = 1/|S| \sum_{F_i \in S} MI(F_i,R) \tag{3.2}$$

In the expressions above, S is the set of features of interest, $MI(.,.)$ is the mutual information function, R is the class variable and F_i is the random variable corresponding to the ith feature. Then the goal of mRMR is to select a feature set S that is as relevant ($\max(Rel)$) and as non redundant ($\min(Red)$) as possible. In the original work [9], two criteria to combine *Rel* and *Red* were proposed. In this work, the criterion of Mutual Information Difference ($MID = Rel - Red$) is used, because it is known to be more stable than the other proposed criterion ($MIQ = Rel/Red$) [16].

As a side note, we have also considered the "feature induction" in [4]. However, we have observed a significant drop in accuracy and therefore will not discuss this approach in this paper.

4. CONDITIONAL RANDOM FIELDS

4.1 Conditional Random Fields

A Conditional Random Field (CRF) is a conditional distribution $p(\mathbf{y}|\mathbf{x})$ in the form of a Gibbs distribution and with an associated graphical structure encoding conditional independence assumptions. Because the model is conditional, dependencies among the input variables x are not explicitly represented, enabling the use of rich and global features of the input (neighboring words, capitalization...). CRFs are undirected graphical models used to calculate conditional probability of realizations of random variables on designated output nodes given the values assigned to other designed input nodes. In the special case, where the output nodes of the graphical model are linked by edges in a linear chain, CRFs make a first-order Markov independence assumption and thus can also be understood as a conditionally-trained finite state machine (FSM).

The distribution related to a given CRF is found using the normalized product of potential functions ($\Psi_C(\mathbf{y}_C)$) for each clique (C). The potential function itself can be, in principle, any non-negative function. Formally, the conditional probability $p(\mathbf{y}|\mathbf{x})$ can be expressed as

$$\begin{aligned} p(\mathbf{y}|\mathbf{x}) &= \frac{1}{Z(\mathbf{x})}\Pi_C\Psi_C(\mathbf{y}_C,\mathbf{x}) \\ &= \frac{1}{Z(\mathbf{x})}exp(-\textstyle\sum_C H_C(\mathbf{y}_C,\mathbf{x})) \end{aligned}$$

On the above equations, $H_C(\mathbf{y}_C,\mathbf{x}) = \log(\Psi_C(\mathbf{y}_C,\mathbf{x}))$. A CRF can also be seen as a weighted finite state transducer [1]. For example, in Figures 4.1 and 4.2, we can see the equivalent expression of a linear chain (1st order) CRF and 2nd order CRF as finite state transducers. These figures clearly show the parameter explosion when the order is increased. Higher number of parameters denies us the possibility of accurate parameter explosion in finite data. Indeed, using CRFs with order greater than one, deteriorates the model performance. On the other hand, a CRF of order 0 discards

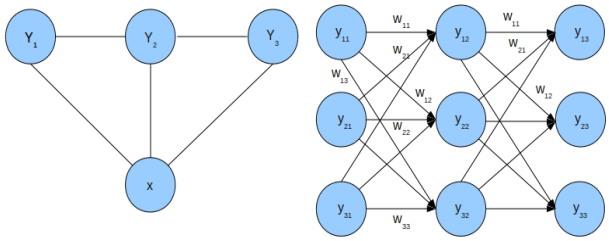

Figure 4.1: The equivalent expression of a linear chain CRF (on the left) as a FST (on the right)

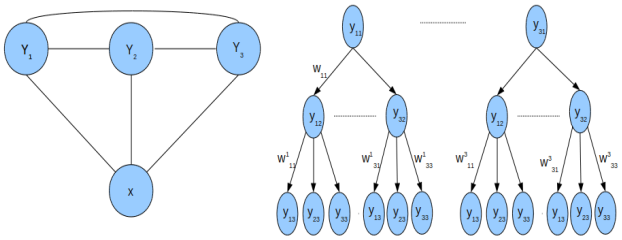

Figure 4.2: The equivalent expression of a 2nd order CRF (on the left) as a FST (on the right)

all neighbourhood information, effectively eliminating the advantages of sequential modeling. Unlike MEMM (see [17]), the transition weights in CRF are unnormalized, the weight of the whole path is normalized instead, which alleviates the label-bias problem.

The associated undirected graph of a CRF also indicates the conditional independence assumptions of the models. In undirected graphs, independence can be established simply by graph separation: if every path from a node in X to a node in Z goes through a node in Y , we conclude that $X \perp Z|Y$. In other words, X and Z are independent given Y. Properly defining conditional independencies is essential in any statistical machine learning application, as having too many parameters will most often result in degraded performance.

4.1.1 Why CRF

CRF is compatible with the nature of problem: CRF is used for computing probability of label combinations of the whole sentence while other methods optimize word by word instead of the whole sentence ,it causes CRF optimize probability better than others, since the results from CRF are more consistent sentence-wise. Robust to over-fitting problem: Since we determine the structure of undirected graph and the structure is fixed(i.e. There is no structural learning involved so the probability of

over-fitting is less) Label Bias problem is solved, early words with low entropy (of $p(w_i|w_{i-1})$) do not cause bias for the solution. We can always find the global optimum of the Likelihood function.CRF is a well known model primarily used in sequential classification problems. We can also think of a CRF as a finite state probabilistic transducer with un-normalized transition probabilities. However, unlike some other weighted finite-state approaches CRFs assign a well-defined probability distribution over possible labellings of a sentence, trained by maximum likelihood, which corresponds to the maximum entropy solution for CRF. Furthermore, the loss(negative log likelihood) function is convex, guaranteeing convergence to the global optimum. CRFs also generalize easily to analogues of stochastic context-free grammars. The transitions leaving a given state compete only against each other, rather than against all other transitions in the model. In probabilistic terms, transition scores are the conditional probabilities of possible next states given the current state and the observation sequence. This per-state normalization of transition scores implies a "conservation of score mass" whereby all the mass that arrives at a state must be distributed among the possible successor states.

In MEMMs, an observation can affect which destination states get the mass, but not how much total mass to pass on. The critical difference between CRFs and MEMMs is that a MEMM uses per-state exponential models for the conditional probabilities of next states given the current state, while a CRF has a single exponential model for the joint probability of the entire sequence of labels given the observation sequence. Therefore, the weights of different features at different states can be traded off against each other. We do not face label bias problem In HMM the structure is not fixed it has a tendency to over-fit A generative model (models p(X,Y)) CRF or MEMM are instead discriminative models (modeling p(Y|X)) Sensitive to marginal distribution of X, thus if X in test data are distributed different to X in training data, the performance suffers. CRF and MEMM are robust to this mismatch.

5. POS-TAGGING USING CRF

5.1 Method

In this section we discussed methods for POS tagging problem and morphological disambiguation problem. In the proposed method, POS tagging of a sentence is performed in a series of steps. In the most basic form we begin by computing the features related to the sentence, later the conditional probabilities of possible tag assignments are computed and the most probable tag sequence are selected. The proposed method makes use of the mallet library [**?**] and the mRMR source code found in [**?**].

5.1.1 Features

In a linear chain conditional random field, there are two types of features: edge features and node features. Edge features are functions of labels of consecutive words ($f_k(y_i, y_{i+1})$) and node features are functions of words in the sentence ($f_k(y_i, \mathbf{x})$, where $\mathbf{x}$ denotes words of the sentence). The probability of a sequence is determined by the feature values as well as the associated model parameters. Thus, determining good feature functions that describe the important characteristics of the words is crucial for a successful model. We employ several morphological/syntactical properties as features.

In our model, the feature functions f_k are determined using several tests such as capitalization, end of sentence, etc. Results of these tests together constitute the features vector $F = f_1, f_2, ..., f_k$ for a word.

To illustrate the two kinds of features, let's consider one feature for node and edge type features used in our model. The *Color* feature is an example for a node feature, it is a function that returns one if the word is among a set of words describing colors and zero otherwise. The indicator function $\Phi(y_i = Adj, y_{i+1} = Noun)$, which returns one if the expression is true and zero otherwise, is an example of an edge feature.

The edge functions in our proposed method consist of all possible slot value pairs. The node functions are given in Table 5.1. The features “Color Set Feature”, “Digit Set Feature”, “Pronoun Set Feature”, “Transition Set Feature” and “Non-Restrictive Set Feature” indicate whether the word is a member of corresponding sets of special words. These sets correspond to specific linguistic classes in Turkish language. The “Noun Adj Feature” indicates whether the word has suffixes that are generally used to change a noun to an adjective. “Capital Feature” indicates whether the word starts with a capital letter. “Before amount feature” and “Before Ques Morpheme Feature” indicate whether the word is followed by a special word/class of words. As their names imply, “Beginning Sentence Feature” and “End Sentence Feature” indicate whether the word is at the beginning or the end of the sentence. Finally, “Equal Slot”, “X2Y Before” and “X2Y After” feature templates generate features based on whether respectively the word itself, the word before or after it has a particular slot value which is unambiguously known, i.e. these values are the same for all possible analyses of the word. We have also considered looking into the previous two and the next two words, but it turned out to degrade the performance. These classes of features contain 363 feature functions. However, in application, some of these features were discarded using mRMR as explained in Section **??**. Figure 5.1 shows a sample sentence and the corresponding features. In this Figure, we observe that the first word "Milosevic’in" gets the "Beginning" feature. Since the morphological analyzer states that the fact that this word is "A3sg", "Noun" and "Prop" unambigously, i.e. these tags showup in all of the possible parses, we also have the "Equal Slot" generated features of "A3sg", "Noun" and "Prop". Finally, we see the feature "X2Y Before A3sg" which means the word after this one is unambigously known to be "A3sg". We can confirm this by checking the next word "kursunu" where we can see the feature "A3sg" as expected. The features for the other words can be understood similarly.

5.1.1.1 Basic Model

The CRF trained for POS tags are conditioned on the features of the sentence. However, during POS tagging, we also know a set of possible tags given by the morphological analyzer, which we call possible solution sequences ($\mathbf{S}_i$). Thus, we have a further conditioning.

Feature Templates	Number of Corresponding Features
Capital_Feature	1
End_Sentence_Feature	1
Begining_Sentence_Feature	1
Color_Set_Feature	1
Equal_Slot_Feature	116
Digit_Set_Feature	1
Before_Mi_Feature	1
Pronoun_Set_Feature	1
Transition_Set_Feature	1
Nonrestrictive_Set_Feature	1
Before_Amount_Feature	1
Noun_Adj_Feature	1
X2Y_Before_slot	116
X2Y_After_slot	116
After_Capital_Feature	1
Proper_Feature	1
PostP_Feature	1
Apostrophe_Feature	1
Total	363

Table 5.1: The features considered in this work

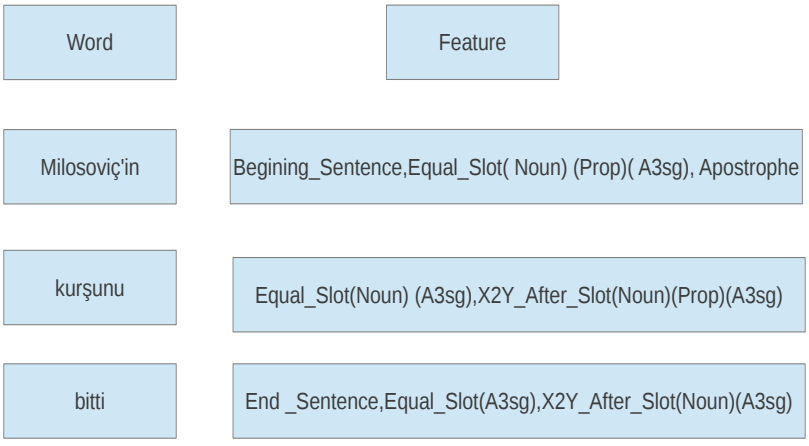

Figure 5.1: A sample sentence and the corresponding features

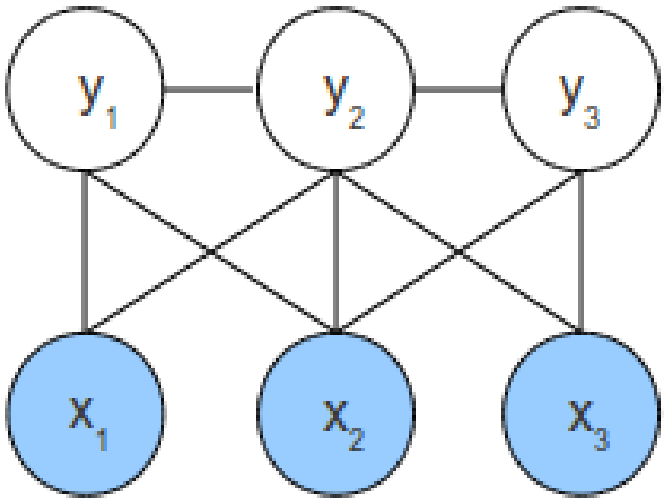

Figure 5.2: The graphical model of the proposed approach

$$p(\mathbf{S}_i|C) = \frac{p(\mathbf{S}_i|\mathscr{F}(C))}{\sum_j p(\mathbf{S}_j|\mathscr{F}(C))} \tag{5.1}$$

Where C is the sentence and $\mathscr{F}(C)$ is the corresponding feature representation of the sentence, given by the CRF. In other words, we do not assign the most probable tag sequence according to the conditional probability given by the CRF but select the most probable sequence ($\hat{\mathbf{t}}$) among possible sequences instead. This selection is performed by a constrained Viterbi approach, where the Viterbi is run on states that are deemed possible by the morphological analyser, instead of running Viterbi on the whole state space.

$$\hat{\mathbf{t}} = \arg\max_{S_i} p(\mathbf{S}_i|C) \tag{5.2}$$

The graphical model for the proposed method is shown in Figure 5.2.

Figure 5.3 shows a sample sentence and how our method chooses the POS tags. The top part of the figure shows the features for the respective words and the bottom part shows the possible POS tags as given by the analyzer. The values indicated above the arrows show transition weights. Note that in this example, any path from a tag of the initial word to a tag of the last word is a possible solution. In this figure, the weights of the transitions are the functions of the initial state, the final state and the features of the final word. The weight function is actually a factored expression, where

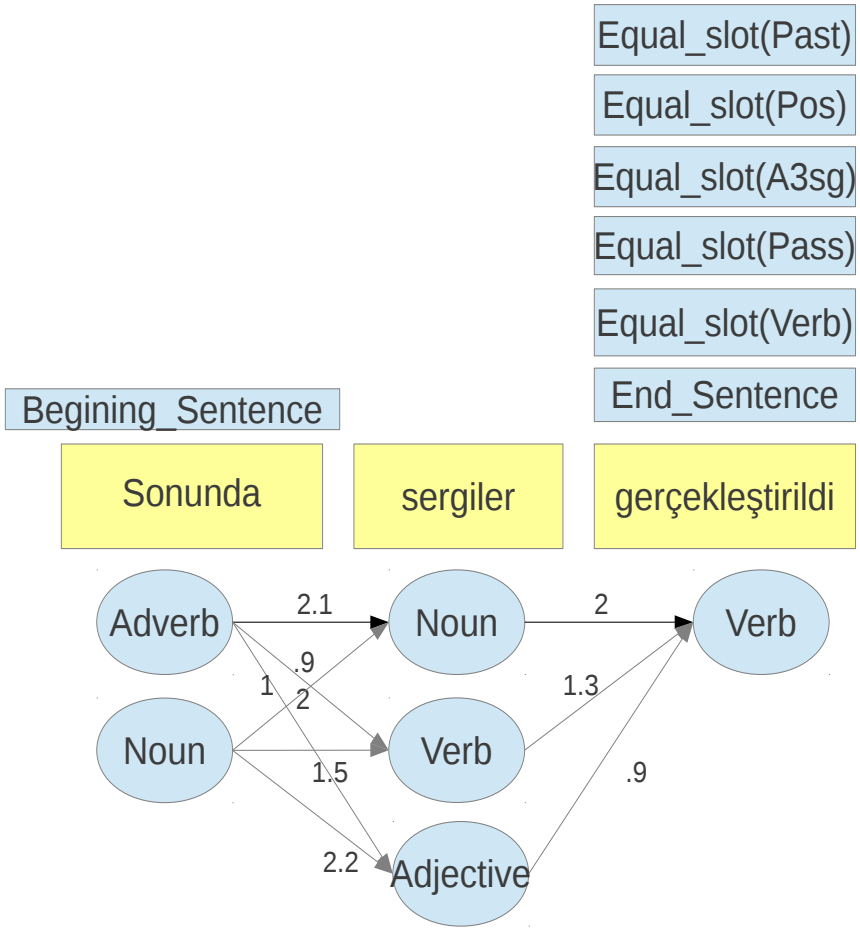

Figure 5.3: A sample sentence ("The exhibition has been finally realized.") with features and possible solutions. The tag chosen by our method is shown in bold arrows.

$f(s_i, s_{i+1}, \mathscr{F}(w_{i+1})) = q(s_i, s_{i+1})q(s_{i+1}, \mathscr{F}(w_{i+1}))$, the first term corresponds to the edge features and the second term corresponds to node features.

5.1.1.2 Alternative Models

The basic approach of using CRF for POS tagging has an important disadvantage: high computational complexity. To remedy this issue, we propose these methods: dividing sentences into shorter sub-sentences and using marginal probabilities of tag assignments per word to eliminate the unlikely tags. In addition, we introduce a new approach to improve the performance of the basic method without significant overhead. In this section, we describe these methods and briefly comment on their performances. The quantitative results will be given in the Results Section.

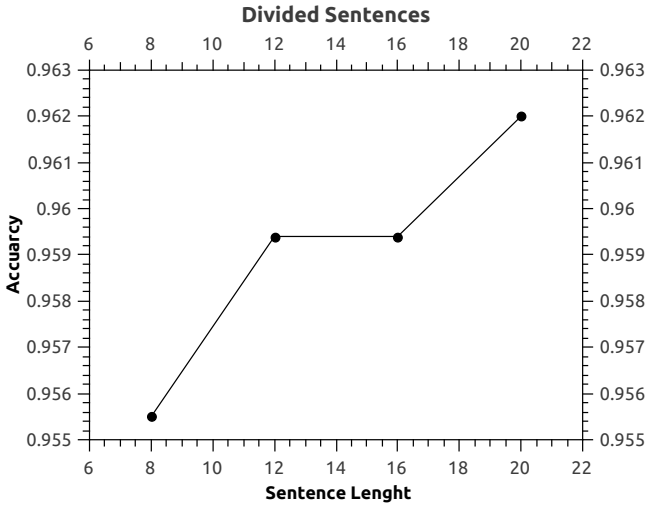

Figure 5.4: Accuracy vs. the length of the partial sentences

Note that the complexity of the constrained Viterbi is $O(T \times |S|^2)$, where T is the length of the sequence and $|S|$ is the maximum number of possible states in any element of the sequence.

5.1.1.3 Model I: Splitting Sentences

This fast approximation method is conceptually the easiest one. The idea is to split a long sentence into multiple parts such that each part is shorter than a maximum length. Let's explain this method with an example sentence from our corpus. This sentence has 35389440 different possible morphological analysis sequences. The poor performance that would result from computing the probabilities of all of these possible solutions is obvious. Now suppose we divide the sentence into 4 parts of lengths 9,9,9,7. The corresponding number of possible solutions are 384, 960, 30 and 32 which sum up to 1406. The huge savings in the number of solutions to consider is apparent. However, despite these good reductions in the number of possible solutions to consider, this method results in the worst accuracy among the alternatives. This is due to the fact that splitting sentences this way enforces an independence assumption on the splitted sub-sentences, which reduces the performance especially in words that are closer to the cut-off boundaries. The Figure 5.4 shows the tradeoff between the performance and the length of the partial sentences.

Using this approach, the complexity of disambiguating a sentence is reduced to $O(T' \times |S|^2)$, where T' is the maximum length of the sub-sentences, so the reduction is linear.

5.1.1.4 Model II: Trim Unlikely Tags

Notice that the compexity of the constrained Viterbi is linear on the length but quadratic on the maximum number of states for any element of the sequence. This observation becomes even more important when we note that the number of possible analysis of a word can reach up to 23 in our corpus and possibly more in general texts. Thus a reduction on the number of possible tag assigments of a word can have significant effects. Out of the many possible sequences for the sentence mentioned in Section 5.2.0.3, many include highly unlikely values for some words. The approach discussed in this section exploits this pattern by trimming out the highly unlikely tags for words but still allowing multiple possible POS tags. In our implementation, we select the words for which the number of possible tag assignments is greater than 6. For such words, we remove the least likely tag assignments using marginal probabilities until either this number is 6 or the number of eliminated tags is 5. We use such an upper limit in order not to remove too many such tags in order not to degrade accuracy. The additional complexity of this approach is obviously linear on the length of the sequence and the trimmed sequence can be disambiguated by constrained Viterbi in $O(T \times 6) = O(T)$. We can see that there can be huge savings in long sentences with compex morphological properties. The conservative approach outlined here means the accuracy is not effected at all, as shown in the next section.

5.1.1.5 Model III: Model Complexity of the Solutions

An interesting observation of morphological properties of words in Turkish is that the correct POS tags of the words tend to be the less morphologically complex ones. In other words, simpler interpretations of words tend to be used more often than the more complex ones of the same word. One way to operationalise this observation is to take the Bayesian stance and model a prior. However, correctly assigning numerical values for our prior knowledge is difficult and we take the other position, where the nature of this relation is learned from the data itself. In Turkish, the morphological complexity of a word can be modeled by the number of IGs of it. Thus we model this number with a 0-order CRF, since we do not expect the neighbouring IG counts to effect each other. This CRF is combined with the original one by multiplying the probabilities, i.e. we

assume the number of IGs and the POS tags to be independent, which is reasonable. Since we use a 0-order CRF, the compexity of inference is only $O(T \times |S|)$. However, we do note increased performance as can be seen in the next section.

5.1.2 Methods for morphological disambiguation step

5.1.2.1 Basic Model

The major problem with trying to estimate $p(\mathbf{y}|\mathscr{F}(S))$ is the enormous number of possible values for $\mathbf{y}$. This big number, together with the available NLP corpora, means the joint estimates for the tags will not be reliable. Our approach is to partition the tags $\mathbf{y}$ into $(y_0, y_1, \ldots, y_9)$ such that each y_0 will take on values from disjoint sets of tags $\mathscr{Y}_i$ and these random variables will be assumed to be conditionally independent, i.e. $p(\mathbf{y}|\mathscr{F}(S)) = \Pi_i p(y_i|\mathscr{F}(S))$. We call the set of possible values $\mathscr{Y}_i$ as slots and the corresponding values as slot values. Sometimes the random variable y_i itself will also be referred as a slot, the distinction will be apparent from the context.

The necessity of assuming a structure for $\mathbf{y}$ is obvious, as otherwise effective estimation will not be possible. However, the particular assumption in our model may still be questioned. However, when we note that the independence assumption is a conditional one, we see that the dependence of slots are still modeled when we take the distribution on input into account. The slots themselves are determined by requiring that no analysis can take multiple different tags from one slot and the slots themselves should have a common semantic interpretation. This can also be seen as another justification of local independence assumption. These two requirements lead us to design the slots as shown in Table 5.2.

Given this slot structure, our approach is to model each slot using a Conditional Random Field. The estimation of model parameters per slot is much less problematic than that of the joint distribution, and we shall demonstrate that this model can disambiguate the morphological tags with a very high success.

5.1.2.2 Improving Efficiency

Slot Groups	Slot Values
Main POS	Adj, Adv, Conj, Det, Dup, Interj, Noun, Num, Postp, Pron, Punc, Verb
Minor POS	Able, Acquire, ActOf, Adamantly, AfterDoingSo, Agt, Almost, As, AsIf, AsLongAs, Become, ByDoingSo, Card, Caus, DemonsP, Dim, Distrib, EverSince, FeelLike, FitFor, FutPart, Hastily, InBetween, Inf, Inf1, Inf2, Inf3, JustLike, Ly, Ness, NotState, Ord, Pass, PastPart, PCAbl, PCAcc, PCDat, PCGen, PCIns, PCNom, Percent, PersP, PresPart, Prop, Quant, QuesP, Range, Ratio,Real, Recip, ReflexP, Rel, Related, Repeat, Since, SinceDoingSo, Start, Stay, Time, When, While, With, Without, Zero
Person Agreements	A1pl, A1sg, A2pl, A2sg, A3pl, A3sg
Possessive Agreements	P1pl, P1sg, P2pl, P2sg, P3pl, P3sg, Pnon
Case Markers	Abl, Acc, Dat, Equ, Gen, Ins, Loc, Nom
Polarity	Neg, Pos
Tense/Mood	Aor, Desr, Fut, Imp, Neces, Opt, Pres, Prog1, Prog2, Cop, Cond, Past, Narr
Compund Tense	Comp_Cond, Comp_Narr, Comp_Past
Cop	Cop

Table 5.2: Morphological Tags

The main concern for the practical application of CRFs in the literature is its efficiency issues. Many authors mention the high complexity of the inference step [12]. In this section, we shall discuss several schemes to improve the efficiency, without reducing the performance as much as possible. The schemes that we will discuss are; dividing sentences, selecting a subset of minimally sufficient identifying tags (distinguishing markers list) and trimming the solution space.

To facilitate comparison, in all of the coming discussion, we shall employ the following sentence as an example (an excerpt from an Omar Khayyam's poem): "Tanrıya toz kondurmamak meleğin işi olsun ve temizlik, cennet kapıcısının işi" (Let the angels try to keep god from blame and the doorkeeper of the heavens do the cleansing). The number of possible morphological analyses for the sentence is 2,1,4,3,4,5,1,3,2,5,4. The number of possible solution sequences is $2 \times 1 \times 4 \times 3 \times 4 \times 5 \times 1 \times 3 \times 2 \times 5 \times 4 = 57600$

5.1.2.3 Dividing Sentences

One of the simplest possible ideas is to split the sentences into non-overlapping subsentences to be disambiguated independently. Of course, since we know the

subsentences are actually parts of a larger sentence, we can still keep the same features for the subsentences as in the original sentence. This splitting procedure can drastically reduce the number of possible sequences. Consider splitting the aforementioned example sentence into subsentences of length at most 4 so that we have subsentences of length 4,4 and 3. The possible solution sequences for each subsentence is 24, 60 and 40, summing up to 124. To determine the probability of each sequence, we have to ask the individual slot probabilities. As there are 9 slots this means we need to evaluate sequences for their probabilities $124 \times 9 = 1116$ times. Compare the figure with $57600 \times 9 = 518400$ of the original problem.

Even though we have a very good reduction in terms of probability evaluations, this method also performs the worst. As we will show in the Experimental Results section, the reduction in performance is too large for this approach to be usable.

5.1.2.4 Distinguishing Markers List

In order to fully disambiguate a sentence we do not need to know the values for all slots. To compute a minimal number of slots to fully disambiguate a sentence, we employ a greedy mechanism. In this approach, iteratively we search for the slot which decreases the ambiguity most when the correct value of the slot is known. We keep adding slots to our distinguishing marker list in this fashion, until there is no ambiguity left.

The ambiguity is measured using entropy. In order to compute the entropy of a sentence, we first compute the entropy of words. The entropy of a word is computed by assuming that the correct analysis is distributed uniformly over the set of possible solutions. The decrease in ambiguity when the correct analysis for slot i is known is calculated using mutual information:

$$\begin{aligned} MI_w(Y_w; Y_w^i | Y_w^S) &= H(Y_w|S) - H(Y_w^| Y_w^i, Y_w^S) \\ &= -log(f_w^S) + log(f_w^{(S \cup i)}) \end{aligned}$$

On the above equation, vector random variable Y_w denotes the correct morphological analysis of word w, while Y_w^j denotes the jth component of the correct analysis. The set S is the DML at the current step. To determine the slot to add to DML, we sum up the $MI_w()$ for all words in the sentence. The slots to be added to DML are determined by:

$\left[s = \arg\max_j \sum_w MI_w(Y_w; Y_w^j | Y_w^{S^t})\right] \left[S^{t+1} = S^t \cup s\right]$ The procedure is repeated until

MI for all remaining slots are 0.

The advantage of determining DML is to increased efficiency of the procedure, as running all 9 CRF classifiers for each sentence is time consuming. The drawback, however, is the degraded performance. One can improve the results by taking an alternative approach, where the slots are added to DML two by two instead of one by one. In this way, the decisions of different classifiers better support each other and higher accuracy is obtained as we will discuss in the Experimantal Results section.

Considering the example sentence, the DML approach gives slots 1,2, and 4 as the minimal distinguishing markers. This reduces the number of probability evaluations of 518400 in the original problem to 172800. As we can see, the reduction is not as impressive as in the previous case.

5.1.2.5 Word-Wise Trimming Unlikely Solutions

Many of the possible sequences for the example sentence include highly unlikely values for some of the words. The approach discussed in this section exploits this pattern by trimming out the highly unlikely tags for words but still allowing multiple possible POS tags. The proposed method reduces the number of possible solution sequences below any given limit. This is achieved using a two step approach. In the first step we trim all words in order to have no more than two possible distinct POS tags, until the number of solutions is below the limit ("trimming limit"). In very long sentences, it is possible that the first step may not reduce the number of possible solutions to the desired level, so we proceed with a more aggressive trimming by selecting the most probable POS tag among the two POS tags offered by the first step for each word, until the number of possible solutions is below the desired trimming limit.

This approach is the most adaptive one, since by keeping the trimming limit high, we have a very good performing method, while on the other hand, keeping the trimming level low (such as 500 or 100) we have a very fast method that still compares well with

other alternatives. If we consider the aforementioned sentence, the first step reduces the number of probability evaluations to $512 \times 9 = 4608$, which may be reduced further using the second step. This approach is the best performing one out of the three approaches when the trimming limit is reasonable (say 5000 or 10000).

6. Experimental Results

6.1 Experimental Results

In this section, we first show the effect of feature selection on the performance. We then show the performance of the proposed method on a common dataset and compare it with the method of [8], which is considered as the state of art. The results are obtained using default parameters of the mallet library. The Java source codes used in the experiments will be made available online.

6.1.1 POS tagging Results

The results for the proposed method, together with the results from [8] (Perceptron) are given in Table 6.1. We use the same training data (1 million words) that is used in these studies. The training data is a semi-automatically tagged data set which consists some erroneous analyses. In this study, we strived to correct as many errors as possible and trained our methods as well as the previous methods on this dataset. We have also accounted to the difference in tags employed in Hasim Sak's method and ours so we kept two separate training files, each having the same corrections but slightly different tags, so that Hasim Sak's method does not suffer from the changes in some of the tag names. Our test data (a manually disambiguated data consisting nearly 1K words) is again from [11]. Note that this set also contains errenous analyses, which we had to correct. All the results are reported using this corrected dataset, which will be made available to researchers. These corrections are the reason why our results are slightly different than the ones reported in [8] The results are reported in Table 6.1.

The results in Table 6.1 exclude the punctuations in computing the accuracy. The results indicate the competitiveness of our approach. It is important to recognize that the POS tagging in Perceptron [8] method is performed by selecting the appropriate tags after a full morphological disambiguation. On the contrary, our method directly

Method	test set
Perc [8]	98.60
Basic Model	98.35
Model I	96.2
Model II	98.35
Model III	98.48
Model II + Model III	98.48

Table 6.1: Pos Tagging Performances

assigns a POS tag sequence to the sentence. The output of our method need not be a single assignment, instead we can output different "belief levels" for different tag assignments. If these POS tags are to be used in another procedure as an intermediate step, this will also be an advantage. Finally, the method in [8] contains a lot more number of features than our proposed approach, since our approach is flexible in the selection features, it can be extended using additional features from the Perceptron method.

6.1.2 Automatic Feature Selection Results

Feature selection is an important step in many machine learning tasks. The effect of feature selection is two-folds, the reduction of features may actually increase classification performance, since accidental correlations in the training data can mislead the classifier and generalization capability of classifiers is expected to be better for lower model complexity. Another effect is the improvement in training and classification efficiency, since inference in the model with a fewer number of features will be faster. For these reasons, we have dismissed the features that are not selected in the top 230 by mRMR.

Figure 6.1 shows the accuracy vs. the number of features. We can see that reducing the features below 230 degrades the performance significantly. Even though a significant increase in performance is not observed for the particular validation set, the reduction in features is still relevant to reduce computational complexity in test and training.

6.1.3 Disambiguation Results

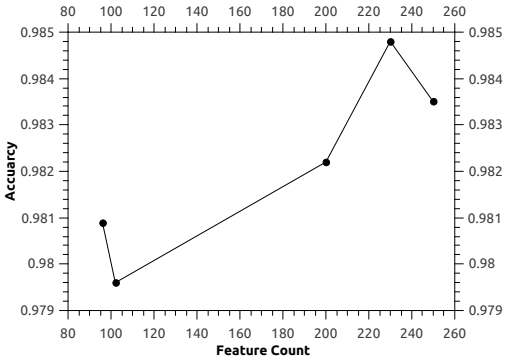

Figure 6.1: Accuracy vs. number of features selected by mRMR

7. CONCLUSIONS

7.1 Conclusions

In this thesis, we proposed a method using Conditional Random Fields to solve the problem of POS tagging and morphologcal disambiguation in Turkish. We have shown that using several features derived from morphological and syntactic properties of words and feature selection, we were able to achieve a performance competitive to the state of art. Furthermore, the probabilistic nature of our method makes it possible for it to be utilized as an intermediate step in another NLP task, such that the belief distribution can be used as a whole instead of a single estimate. Note that our proposed method can also be employed to other languages, perhaps with the addition of language dependent features.

Another major contribution of this work is the discussion on several approaches to improve efficiency of POS tagging using CRFs. We believe this work constitutes a major step towards making CRF a more practical tool in NLP.

As part of our future work, we plan to investigate the addition of other features to improve the performance of the proposed method. One possibility is to incorporate features based on lemma. Eventually, we plan to combine several CRF models to solve the full disambiguation task, which poses several interesting challenges.

REFERENCES

[1] **Smith, N.A.**, **Smith, D.A. and Tromble, R.W.**, 2005. Context-based morphological disambiguation with random fields, Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, HLT '05, Association for Computational Linguistics, Stroudsburg, PA, USA, pp.475–482, http://dx.doi.org/10.3115/1220575.1220635.

[2] **Sutton, C. and McCallum, A.**, 2010. An introduction to conditional random fields, *Arxiv preprint arXiv:1011.4088*.

[3] **Lafferty, J.**, **McCallum, A. and Pereira, F.C.N.**, 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

[4] **McCallum, A. and Li, W.**, 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons, Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4, CONLL '03, Association for Computational Linguistics, Stroudsburg, PA, USA, pp.188–191, http://dx.doi.org/10.3115/1119176.1119206.

[5] **Kudo, T.**, **Yamamoto, K. and Matsumoto, Y.**, 2004. Applying conditional random fields to Japanese morphological analysis, Proc. of EMNLP, volume2004.

[6] **Shacham, D. and Wintner, S.**, 2007. Morphological disambiguation of Hebrew: A case study in classifier combination, Proceedings of EMNLP-CoNLL, volume 7, pp.439–447.

[7] **Habash, N. and Rambow, O.**, 2005. Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop, Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, ACL '05, Association for Computational Linguistics, Stroudsburg, PA, USA, pp.573–580, http://dx.doi.org/10.3115/1219840.1219911.

[8] **Sak, H.**, **Gungor, T. and Saraclar, M.**, 2007. Morphological Disambiguation of Turkish Text with Perceptron Algorithm, Proceedings of the 8th International Conference on Computational Linguistics and Intelligent Text Processing, CICLing '07, Springer-Verlag, Berlin, Heidelberg, pp.107–118, http://dx.doi.org/10.1007/978-3-540-70939-8_10.

[9] **Peng, H.**, **Long, F. and Ding, C.**, 2005. Feature selection based on mutual information criteria of max-dependency, max-relevance, and

min-redundancy, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **27(8)**, 1226 –1238.

[10] **Hakkani-Tür, D.**, **Oflazer, K. and Tür, G.**, 2002. Statistical Morphological Disambiguation for Agglutinative Languages, *Computers and the Humanities*, **36(4)**, 381–410, http://www.springerlink.com/content/p8217051382887pm/abstract/.

[11] **Yuret, D. and Töre, F.**, 2006. Learning morphological disambiguation rules for Turkish, Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, HLT-NAACL '06, Association for Computational Linguistics, Stroudsburg, PA, USA, pp.328–334, http://dx.doi.org/10.3115/1220835.1220877.

[12] **Sutton, C.**, **Mccallum, A.**, **Getoor, L. and Taskar, B.**, 2006. Introduction to Conditional Random Fields for Relational Learning, Introduction to Statistical Relational Learning, MIT Press.

[13] **Arslan, B.B.**, 2009, An Approach To The Morphological Disambiguation Prolem Using Conditional Random Fields.

[14] **Oflazer, K.**, 1995. Two-level Description of Turkish Morphology, *Literary and Linguistic Computing*, **9(2)**, 137–148, http://llc.oxfordjournals.org/content/9/2/137.full.pdf+html.

[15] **Guyon, I. and Elisseeff, A.**, 2003. An introduction to variable and feature selection, *J. Mach. Learn. Res.*, **3**, 1157–1182, http://dl.acm.org/citation.cfm?id=944919.944968.

[16] **Gulgezen, G.**, **Cataltepe, Z. and Yu, L.**, 2009. Stable and Accurate Feature Selection, **W. Buntine**, **M. Grobelnik**, **D. Mladenić and J. Shawe-Taylor**, editors, Machine Learning and Knowledge Discovery in Databases, volume5781, chapter 47, Springer Berlin Heidelberg, Berlin, Heidelberg, pp.455–468, http://dx.doi.org/10.1007/978-3-642-04180-8_47.

[17] **McCallum, A.**, **Freitag, D. and Pereira, F.C.N.**, 2000. Maximum Entropy Markov Models for Information Extraction and Segmentation, Proceedings of the Seventeenth International Conference on Machine Learning, ICML '00, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp.591–598, http://dl.acm.org/citation.cfm?id=645529.658277.

CURRICULUM VITAE

Candidate's full name:	Razieh Ehsani
Place and date of birth:	Shabestar,Iran, 09 December 1984
Universities and Colleges attended	Istanbul Technical University
Address:	ITU Ayazaga Campus, Faculty of Computer and Informatics, Sarıyer, İstanbul
Phone:	0212 5029882
email:	rehsani@itu.edu.tr