ARABIC PART OF SPEECH DISAMBIGUATION: A SUPERVISED STOCHASTIC MORPHEME-BASED APPROACH

MOHAMMED YAHYA ALI ALBARED

THESIS SUBMITTED IN FULFILMENT FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

FACULTY OF INFORMATION SCIENCE AND TECHNOLOGY
UNIVERSITI KEBANGSAAN MALAYSIA
BANGI

2012

PENYAHKABURAN KELAS KATA BAHASA ARAB: PENDEKATAN BERASASKAN MORFEM STOKASTIK DISELIA

MOHAMMED YAHYA ALI ALBARED

TESIS YANG DIKEMUKAKAN UNTUK MEMPEROLEH IJAZAH
DOKTOR FALSAFAH

FAKULTI TEKNOLOGI DAN SAINS MAKLUMAT
UNIVERSITI KEBANGSAAN MALAYSIA
BANGI

2012

DECLARATION

I hereby declare that the work in this thesis is my own except for quotations and summaries which have been duly acknowledged.

16 JULY 2011

MOHAMMED YAHYA ALI ALBARED
P45081

AKNOWLEDGEMENT

First and for most, I am thankful to Almighty Allah for making me able to work on and complete this thesis. After that, I owe a great debt of gratitude to my supervisor, Associate Professor Dr. Nazlia Omar for her guidance and patience throughout the completion of this research. This thesis would not have been possible without your guidance, as well as inspiring and enlightening ideas, comments and suggestions. I am glad to be your supervisee and share your rich, broad and deep knowledge. Thank you very much. Thank you also for your concern, your endless support and readiness to help me whenever I needed. You made me comfortable to come to you whenever I have problems.

I would also like to express my sincere appreciation to my second supervisor, Associate Professor Dr. Mohd. Juzaiddin Ab Aziz, for his guidance and patience throughout the completion of this research. Thank you for your critical insights, constructive comments and shared experience. Thank you very much.

I would like to thank the entire I would also like to thank the entire staff of the Faculty of Information Science and Technology at the University of Kebangsaan Malaysia particularly Prof. Dr. Shahrul Azman Mohd Noah, our research group Knowledge Technology (KT) leader, for his support and shared experience. All this support has allowed me to concentrate my efforts on my research. In addition, a big thank to all KT group members for valuable discussions and to share their knowledge and professional experience.

Last, and the most, I would like to express my appreciation to my family for their love and care without which the completion of this program would not have been possible. My heartfelt thanks go to all my family members: my father, my mother, my wife, and my lovely children whose sacrifice, support, love, caring inspired me to overcome all the difficulties throughout my entire PhD program.

ARABIC PART OF SPEECH DISAMBIGUATION: A SUPERVISED STOCHASTIC MORPHEME-BASED APPROACH

ABSTRACT

Part of Speech (POS) disambiguation is the ability to computationally determine which POS of a word is activated by its use in a particular context. Arabic is a highly inflectional and morphologically rich language, which presents several challenges for POS tagging such as ambiguity and data sparseness, large existence of unknown words and fine-grained and large tag sets. Most POS tagging algorithms are either rule-based or stochastic. While rule-based methods require a large effort, stochastic taggers methods require large annotated corpora for each genre. The creation of such corpora is time consuming and labor intensive. With the lack of such large corpora, this dissertation describes the investigations we carried out in order to find out the best strategy to develop efficient and robust Arabic POS and morphological tagging models that require a small amount of tagged corpora. The baseline tagging models are based on a Hidden Markov Model (HMM), namely, Bigram HMM and Trigram HMM are investigated. Several dynamic smoothing techniques are used with HMM models to overcome the data sparseness problem. This research also presents new methodologies to manage the problem of unknown word POS tagging in Arabic. Firstly, this work designs, implements and empirically evaluates several language independent lexical models based on word affixes probabilities. Secondly, new statistical integrated tagging models are introduced to provide adaptive and transportable tagging scheme. Thirdly, to deal with non-concatenative nature of Arabic word, a new statistical light-pattern based unknown word handler is introduced. This work also studies the influence of the tokenization level on the tagging performance of the tagging models, in term of accuracy and time complexity, in order to determine the best tokenization choice for POS tagging. Finally, to deal with more fine-grained POS tag set, this work presents the morpheme-based Arabic morphological disambiguator, which consists of several morpheme-based N-attributes stochastic classifiers and a module which combines them. Two Arabic small corpora from different genres are used for evaluation, the FUS-HA corpus and the Quranic Arabic Corpus. The best POS tagging result achieved by the new tagging models is 94.5% for unknown word tagging and 96.5% for the overall tagging. Experimental results also show that the new tagging models significantly improve the overall tagging accuracy over the baseline models and perform better than existing Arabic systems on 15 test sets from 15 different genres. In addition, results show that morpheme-based tagging models are more efficient and accurate than word-based models. Finally, our N-attributes stochastic classifier combination model provides morphological tagging with overall accuracy of 91.5% and saves run time over the direct classification approaches.

PENYAHKABURAN KELAS KATA BAHASA ARAB: PENDEKATAN BERASASKAN MORFEM STOKASTIK DISELIA

ABSTRAK

Penyahkaburan kelas kata adalah kebolehan untuk menentukan kelas kata secara berkomputer akan sesuatu perkataan menurut konteks penggunaannya. Bahasa Arab ialah bahasa yang banyak imbuhan dan bahasa yang cukup kaya dengan morfologi, yang berhadapan dengan pelbagai cabaran dalam penandaan kelas kata seperti kekaburan dan kejarangan data, kewujudan perkataan tak diketahui yang banyak dan keperincian set tanda yang banyak. Kebanyakan algoritma-algoritma penandaan kelas kata adalah sama ada berasaskan petua atau stokastik. Bagi kaedah berasaskan petua ianya memerlukan pengembelingan usaha yang banyak manakala kaedah penanda stokastik pula memerlukan kesediaan korpus beranotasi yang cukup besar bagi setiap jenisnya. Pengujudan korpus tersebut mengambil masa yang panjang dan tenaga kerja yang intensif. Dengan kekurangan korpus tersebut, disertasi ini menghuraikan kajian yang kami jalankan bagi mencari strategi terbaik untuk membangunkan model penanda kelas kata bahasa Arab yang efisyen dan teauh dengan hanya menggunakan jumlah bahan latihan sedia ada yang terhad. Model yang menjadi penanda aras ialah Hidden Markov Model (HMM) iaitu HMM Dwigram dan HMM Trigram telah diselidiki. Pelbagai teknik pelicinan digunakan terhadap model HMM untuk mengatasi masalah kejarangan data. Penyelidikan ini juga mengemukakan metodologi baharu untuk mengendalikan masalah penandan kelas kata perkataan yang tidak diketahui bagi bahasa Arab. Pertamanya, kerja ini melibatkan rekabentuk, implimentasi dan penilaian empirikal model leksikal tak bersandar bahasa yakni kebarangkalian imbuhan perkataan. Keduanya, model penandaan bersepadu statistik yang baharu diperkenalkan untuk menyediakan skema penandaan yang adaptif dan boleh angkut. Ketiganya, untuk berhadapan dengan perkataan Arab yang secara semulajadinya tak konkatinatif, maka diperkenalkan pengendali perkataan tak diketahui berasaskan corak-ringkas statistikal yang baharu. Penyelidikan ini juga mengkaji pengaruh tahap pentokenan ke atas keupayaan penandaan bagi model tersebut dari segi kejituan dan kekompleksan untuk tujuan menentukan pilihan pentokenan terbaik dalam penandaan kelas kata. Akhirnya, untuk berhadapan dengan set kelas kata yang lebih terperinci, penyelidikan ini mengemukakan penyahkabuaran morfologi berasaskan morfem Arab yang mengandungi pelbagai pengkelasan stokastik N-sifat berasaskan morfem Arab dan satu modul yang menggabungkan pengkelasan tersebut. Ada dua korpus kecil dari jenis yang berbeza telah digunakan untuk penilaian iaitu korpus FUS-HA dan korpus Al-Quran. Keputusan penandaan kelas kata yang terbaik yang dicapai dengan menggunakan model panandaan baharu adalah 94.5% bagi penandaan perkataan tak diketahui dan 96.5% bagi penandaan keseluruhan. Keputusan eksperimen juga menunjukkan bahawa kejituan penandaan keseluruhan dengan model penandaan yang baharu adalah meningkat dengan jelas berbanding model garis asas dan berkeupayaan lebih baik dari kaedah ke atas bahasa Arab sedia ada bagi 15 set ujian dari 15 jenis yang berbeza. Lantaran itu, keputusan menunjukkan bahawa model penandaan berasaskan morfem adalah lebih effisyen dan tepat daripada model berasaskan perkataan. Akhir sekali, medel penggabungan pengkelasan stokastik N-sifat menyediakan penandaan morfologi dengan kejituan

keseluruhan sebanyak 91.5% dan menjimatkan masa larian berbanding dengan pendekatan pengkelasan langsung.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BI-KN-S	Arabic Bigram tagger with Kneser Ney smoothing
BI-LP-S	Arabic Bigram tagger with Laplace smoothing
BI-MKN-S	Arabic Bigram tagger with Modified Kneser Ney smoothing
BI_WB_S	Arabic Bigram tagger with Witten Bell smoothing
AMT	Arabic Morphosyntactic Tagger
ATB	Arabic TreeBank
ATT+MA+ACF+SP	Integrated tagging model with Augmented Characters Form's affixes
ATHMM+P	Arabic Trigram HMM tagger with Prefix tries
ATHMM+PrS	Arabic Trigram HMM tagger with Pruned Suffix tries
ATHMM+LIG	Arabic Trigram HMM tagger with the linear interpolation guessing
ATT+ACF+SP	Arabic Trigram HMM Tagger with the linear interpolation of both Augmented Characters Form's affixes
AC	Augmented Characters
ACF	Augmented Characters Forms
BP	Broken Plural
BAMA	Buckwalter Arabic Morphological Analyzer
CA	Classical Arabic
CGC	Computational Grammar Coder
CRF	Conditional random fields
CE	Contrastive Estimation
CCA	Corpus of Contemporary Arabic
CDN	Cyclic Dependency Network
ENGCG	ENGlish Constraint Grammar
EM	Expectation Maximization
HMM	Hidden Markov Model
HMC	Human–Machine Communication
THHM+MAP	Integrated tagging model with the proportional weighting function
THHM+MAU	Integrated tagging model with uniform weighting function
KL	Kullback-Leibler
MA	Morphological Analyzer
ML	Machine Learning
ME	Maximum Entropy
MLE	Maximum Likelihood Estimation

MBL	Memory Based Learning
MSA	Modern Standard Arabic
NLP	Natural Language Processing
POS	Part of Speech
QAC	Quranic Arabic Corpus
RNN	Recurrent Neural Network
RSE	Rule-based Stemming Engine
SVD	Singular Value Decomposition
SVM	Support Vector Machine
TDIDT	Top Down Induction of Decision Tree
TBL	Transformation Based Approach
TnT	Trigrams'n'Tags Tagger
TRE	Trilateral Root Extraction
UTBL	Unsupervised Transformation Based Learner

CHAPTER I

INTRODUCTION

1.1 INTRODUCTION

Natural Language Processing (NLP) can be defined as an area of research and application that explores how computers can be used to perform useful tasks involving human language to enable Human–Machine Communication (HMC) , to improve human-human communication or simply to do useful processing of text or speech (Chowdhury 2003; Jurafsky et al. 2009). NLP generally involves six phases including phonetics and phonological analysis, morphological analysis, syntactic analysis, semantic analysis, pragmatic analysis and discourse integration(Allen 1995; Jurafsky et al. 2009).

Part Of Speech (POS) disambiguation is the ability to computationally determine which POS of a word is activated by its use in a particular context. It also can be defined as the process of assigning an appropriate POS tag for each word in a sentence. Fine-grained POS (morpho-syntactic or morphological) tagging is the process of assigning POS , tense, number, gender, and other morphological information to each word in a sentence (Feldman 2006; Schmid & Laws 2008). POS tagging is a necessary fundamental language analysis tasks in most, not to say, all NLP systems such as corpus annotation projects, information extraction, word-sense disambiguation, and many other tasks. The output of POS taggers is usually forwarded to another high-level language analysis task such as named entity recognition (Benajiba 2009)and syntactic parsing (Mohamed 2010).

Research on POS tagging has a long history. Research in automated POS tagging was firstly started in middle sixties and seventies (Harris 1962; Klein & Simmons 1963; Greene & Rubin 1971). Numerous approaches have been successfully applied to POS tagging. These approaches can be classified into two main groups according to the nature of knowledge they use: linguistic and Machine Learning (ML) family. Linguistic-based taggers represent the knowledge involved as a set of rules, or constraints, written by linguists (Mārquez 1999; Màrquez et al. 2000; Loftsson 2008). The linguistic models range from few hundred to several thousand rules, and they usually need years of labor. Researchers manually designed rules for tagging. On the other hand, most of the new approaches stem from the field of ML. From a ML viewpoint, POS disambiguation can also be viewed as a classification problem: the tag set are the classes and an automatic classification method is used to assign each occurrence of a word to one class based on the evidence from the context. These ML methods range from fully unsupervised methods to methods with full supervision.

In recent years, several unsupervised ML methods are used for POS tagging(Smith 2006; Gao & Johnson 2008; Snyder et al. 2008; Gael et al. 2009; Abend et al. 2010). They eliminate the need of manual annotated data by extracting the required information from raw text. Unsupervised learning is significantly harder and less accurate than supervised learning (Wang & Schuurmans 2005). Supervised approaches proved to be more accurate than other approaches given a huge amount of manually tagged corpora(Chenda & Kameyama 2007b; Navigli 2009). Given a poor resource scenario, all ML approaches performs poorly and their tagging accuracies are dropped substantially. The problem of the lack of language resources, i.e. annotated training corpora, is a general problem even for well-studied languages (Marques & Lopes 2001). The available huge amounts of pre-tagged texts are only suitable for some domains. It well known that taggers trained with the existing hand tagged corpora perform quite poorly in new domains due to unknown words problems(Marques & Lopes 2001; Smith et al. 2004; Tsuruoka et al. 2005; Giesbrecht & Evert 2009). All this stress the importance of designing accurate taggers which trained only on small tagged corpora.

This work describes extended supervised stochastic tagging models based on Hidden Markov Model (HMM) with a small amount of tagged corpus for both POS and morpho-syntactic tagging of Arabic. There are many reasons which stimulate researchers to use supervised stochastic framework based on HMM. First, HMM tagging models are efficient in terms of accuracy. Several HMM tagging models are among the most successful supervised POS tagging approaches such as TnT tagger (Brants 2000). In terms of time efficiency, HMM tagging models are extremely fast compared to the other approaches (Megyesi 2001a; Sjöbergh 2003; Giesbrecht 2008). Second, another very important positive aspect, especially with small training data, is that HMM tagging modeling is easily extendable and tunable. A new smoothing algorithm or a new unknown word handler algorithm can be easily integrated with the general framework to develop more accurate model (Padró & Padró 2004; Agic et al. 2008; Chang et al. 2010). Finally, most of the successful morphosyntactic (fine-grained POS) tagging models are based on extended stochastic (HMM) models (Schmid & Laws 2008; Görgün & Yildiz 2011; Schwartz et al. 2011; McClanahan 2010).

1.2 THE PART-OF-SPEECH TAGGING PROBLEM

Part-of-speech (POS) tagging is an essential and well-known NLP problem. POS tagging involves many difficult problems, such as inherent POS ambiguities, and (most seriously) many types of unknown words. The next subsections will discuss the general problem in the POS tagging problems:

1.2.1 Ambiguity Problem

Natural languages are inherently ambiguous in their nature (Tomita 1985; Kumar et al. 2010). Ambiguity appears at different levels of the natural language processing (NLP) task (Dandapat 2009; Jurafsky et al. 2009). If the ambiguity appears in one word it is called lexical ambiguity such as POS ambiguity(Manning & Schutze 1999) (Manning & Schutze 1999). If the ambiguity takes place in a sentence or clause level, it is called the structural ambiguity such as prepositional phrase attachments (Volk

2001; Jurafsky et al. 2009). Consider, for instance, the following English and Arabic sentences:

1. That round table might collapse (Nugues 2006).
2. حسن أكبر من صالح و اكرم *" Hassan is older than Saleh and Akram"*

Each sentence has lot of POS ambiguity which should be resolved before the sentence can be understood or processed i.e. forwarded to higher level of NLP analysis. Figure 1.1 shows a complete analysis of the tagging ambiguity of the English sentence. The colored boxes indicate the correct POS tag of a word form a set of possible tags.

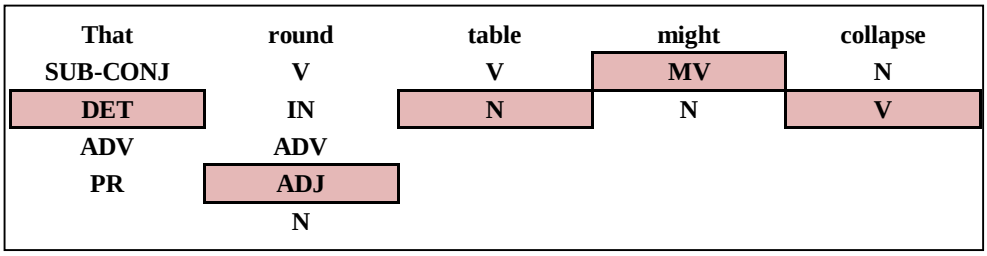

FIGURE 1.1 POS ambiguity of an English sentence
Source: Nugues 2006

In most cases POS ambiguity can be resolved by examining the context of the surrounding words. However, Figure 1.2 illustrates the detail of the ambiguity class for the Arabic sentence as per one of the tag sets used in this work.

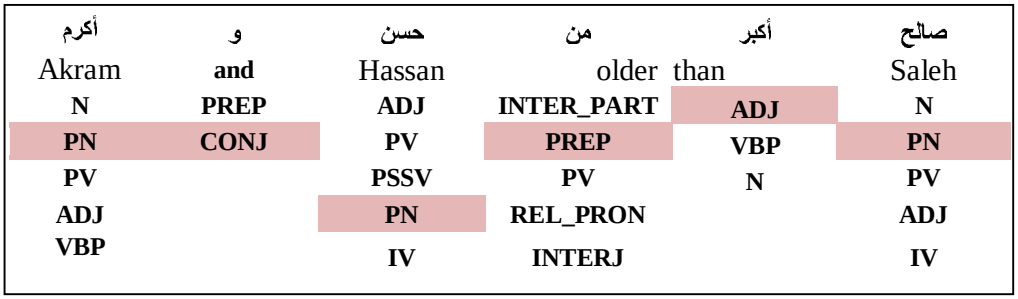

FIGURE 1.2 POS ambiguity of an Arabic sentence

However, Arabic, like other Semitic languages, has a rich and complex inflectional, derivational and templatic morphology(Hajič et al. 2005; Smrz 2007; Hijjawi et al. 2011). For such morphologically-rich languages, the POS tag set should be more fine-grained and defined in terms of morphological and grammatical Features characterizing word structure including typical POS tags, person, number, gender, case, mood, etc (Atwell 2008). In this case, the POS tagging problem is called fine-grained POS tagging, morphosyntactic disambiguation or morphological disambiguation (Halteren 1999; Halteren et al. 2001; Schmid & Laws 2008). The main important qualitative distinction between the typical POS tagging in simple languages and the fine-grained POS tagging (morphological disambiguation) is the large number of possible tags that can be assigned to a word (Levinger et al. 1995; Yuret & Türe 2006; Adler 2007). The level of ambiguity is higher in the fine-grained POS tagging than in the typical POS tagging, because it include two types of ambiguities: ambiguity between POS classes and ambiguity within the same POS class. For example, the token"حسن" *"Hassan"* has five possible typical POS classes as shown in Figure 1.2, and nine fine-grained POS classes as shown in Table 1.1.

Table 1.1 The Possible Fine-Grained POS Tags of the Word "حسن""Hassan".

WORD	TAG
حسن	PN._._.DEF._._._.
حسن	ADJ._.M.DEF._._._.
حسن	ADJ._.M.INDEF._._._.
حسن	N._.M.DEF._._._.
حسن	N._.M.DEF.P._._.
حسن	N._.M.INDEF._._._.
حسن	V.3.M._.S.PERF.ACT.
حسن	V.2.M._.S.IMPV.ACT.
حسن	V.2.F._.S.IMPV.ACT.
حسن	V.3.M._.S.PERF.PASS.

1.2.2 Unknown Words Problem

The term "Unknown Words" denotes words which are not found in neither a training corpus nor a dictionary. The unknown words play an important role in the meaning of a sentence more than known words. Unknown words are specialized

words and hold more semantic information than known word (Vadas & Curran 2005). They also belong to open POS classes such as nouns and verbs and unlikely to be in the closed classes such as particles. In real life scenario, sources of open-ended text such as web corpus present NLP systems with major challenge unknown words(Weischedel et al. 1993). Actually, the size of this problem is proportional to many factors such as size, genre and the quality of the training data(Chenda & Kameyama 2007a).

In any POS tagging model, a special mechanism to guess unknown words POS tags is needed; no matter in which method a tagger is implemented (Lu 2006). Unknown words handling is an important key to improve the performance of the POS tagger (Hall 2003). A POS tagging model which better handles the unknown words is considered to be a well fitted model for the POS disambiguation task in a poor resource scenario (Dandapat 2009).

1.2.3 Is Part-Of-Speech Tagging A Solved Task?

Part-of-speech (POS) tagging for well-studied languages such as English and German is often considered a solved problem. For instance, there is a dozen of free accessible open sources tagging tools for English. In addition, there is a dozen of diverse in topic or writing styles training corpora free accessible to the reaserch community. Morever, there are hundreds of research, started in middle sixties and still continue until now, has been conducted to tackle this problem. Taking these facts into account one may think that POS tagging is a solved and closed problem being this accuracy perfectly acceptable for most NLP applications.

The question i.e. "is POS tagging a solved task and why should we continue to work on the POS tagging problem?" has been raised recently by many researchers (Peshkin et al. 2003; Curran et al. 2006; Giesbrecht 2008; Giesbrecht & Evert 2009; Kübler et al. 2010; Manning 2011). According to them, POS tagging is not a solved problem due to many reasons. First, even a small percentage of errors($\approx$3%) may derail subsequent high level processing steps such as parsing. Current state of the art taggers which demonstrate high word-level accuracy of $\approx$97% have only sentence-

level accuracies around 55–57% , which is a much more modest result(Manning 2011). These tagging accuracies also go down markedly when there are differences between the training and testing data. Second, the performances of these taggers are not robust if a large proportion of words are unknown, or if the testing corpus varies in topic, epoch, or writing style from the training corpus (Kübler et al. 2010). According to Giesbrecht and Evert (2009), the tagging accuracies of 97%–98% are optimistic estimates representing an ideal case for ML approaches. In a real-life scenario, tagging accuracies drop below 93%, making the taggers unsuitable for fully automatic processing".

This is the case, of course, when the language is well-studied and resource-rich like English, German and French. As a matter of fact, in case of less-studied and resource-poor languages like Arabic, the problem is far from being solved. In addition, as a morphologically-rich language, Arabic POS tagging i.e. the morphosyntactic disambiguation is much harder than typical POS tagging in simple language. Arabic morphosyntactic disambiguation has not been studied extensively, even in comparison with languages with morphology quite similar to Arabic like Hebrew. Difficulties and problems of the Arabic POS tagging will be discussed in the next section.

1.3 THE PROBLEM STATEMENT

Most POS tagging algorithms are either rule-based or stochastic. Handcrafting a set of rules requires a large effort. On the other hand, stochastic taggers require a huge manually annotated corpus and similar in style and genre to the test data. The creation of such data is time-consuming and labour-intensive (Umansky-Pesin et al. 2010). Up to date, the work done in the area of POS tagging dedicated to Arabic is quite small due to the lack of such publicly available large corpora (Duh & Kirchhoff 2006; Maamouri et al. 2008; Sawalha & Atwell 2009a; Sawalha & Atwell 2011). It is also hard to prepare such corpora for all text and task type pairs (Kazama 2001; Barrett & Weber-Jahnke 2011). It is well known that all supervised models perform poorly when they are trained using small data. So the main problem here is to find efficient typical and fine-grained POS tagging methods that require a small amount of tagged

corpus. In addition, the Arabic language has some characteristics which harden the POS tagging task. Therefore, any new tagging models should take in account the Arabic characteristics. However, to better handle Arabic POS problem, the research should address several issues. The first issue is that Arabic is highly ambiguous (Zibri et al. 2006; Attia 2008; Al-Taani & Abu Al-Rub 2009; Altabbaa et al. 2010).The main problem here is to determine how stochastic POS tagging models perform with Arabic text and which one is more appropriate for handling the Arabic POS ambiguity given small amount of data. The second issue is the data sparseness caused by the agglutinative characteristic of Arabic words and the small training data. The problem here is to find out what is the impact of incorporating advance smoothing techniques to handle data sparseness problem on the overall tagging results. The third issue is related to the large existence of unknown words, one of the main and the most challenging problems in POS tagging (Lu 2006; Nakagawa 2006; Subramanya et al. 2010; Zeng & Curtis 2010). Their existence always results in dramatically degradation of the overall tagging accuracy. The main problem here is to find out how existing and efficient methods perform in Arabic and also how can Arabic linguistic be utilized to come up with new efficient statistical models for Arabic unknown word POS tagging. The fourth issue is to determine which is the appropriate tokenization level (morpheme or word) that should be used in Arabic POS tagging. Finally, the last issue is about the fine-grained (morpho-syntactic) tagging complexity where the ambiguity is higher and the data sparseness is much harder (the tag set is large). So the researcher has to find how to design a new innovative stochastic paradigm to handle Arabic fine-grained POS disambiguation.

1.4 RESEARCH OBJECTIVES

The primary goal of the research is to design and implement efficient and robust statistical Arabic POS tagging models when small amount of tagged data are available. To address this broad objective, we identify the following goals:

- To design and implement stochastic POS tagging models for Arabic languages based on the investigation of different configurations of the HMM models and smoothing algorithms.

- To extend the stochastic tagging models by designing new lexical models for handling unknown words POS guessing.
- To design and implement a new stochastic paradigm for Arabic morphosyntactic (fine-grained POS) disambiguation based on the combination of several N-attributes stochastic classifiers.
- To evaluate the performance of the proposed tagging models and to compare it with existing systems.

1.5 MOTIVATIONS

POS analysis and disambiguation task is an essential component in many NLP applications including – speech synthesis and recognition, information extraction, partial parsing, machine translation, lexicography etc.

Arabic language is a Semitic language spoken by over 250 million people. It is one of the seven official languages of the United Nations. Unfortunately, there is no an open source available POS tagger, which is designed and developed especially for Arabic to address the community's need for fundamental NLP tools. In addition, due to the complexity of the Arabic POS disambiguation problems, and the limitations of the current work in the literature, thus, the Arabic POS disambiguation problems still need more investigations.

To date, there has been little research in the area of statistical NLP for Arabic, which is hindered by the lack of publicly available manually annotated corpora. In order to reduce the huge cost of manually creating annotated corpora, the development of the POS taggers is of supreme importance.

1.6 THE RESEARCH METHODOLOGY

The research methodology employed by this research is based on the experimental approach where prototypes are developed as a proof of concept based on selected

models and then evaluated using a set of experimental data(corpora) and compared to existing implementations where performance results are available.However, the methodology employed by this research, as shown in Figure 1.3, covers the following:

a) Corpus planning and compiling phase
b) Feasible initial tagging models design and implementation
c) Tagging models improvement phases.
d) Evaluation phase.

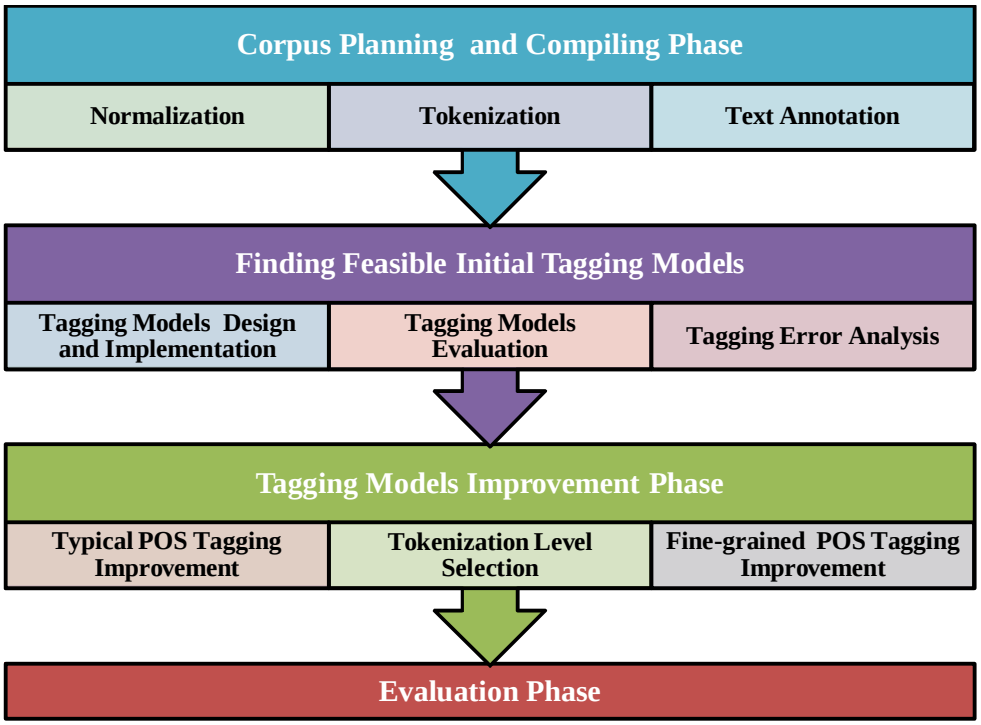

FIGURE 1.3 The Proposed Methodology Architecture

Corpus planning and compiling phase: The main input of this phase is a raw text from both Classical Arabic (CA) and Modern Standard Arabic (MSA). The main output of this phase is the FUS-HA corpus, an annotated linguistic resource which shows the Arabic POS for each word. In contrast with other Arabic annotated corpora, the raw text of FUS-HA corpus is coming from both MSA and CA. In fact, The FUS-HA corpus is primarily intended for training statistical models for POS tagging of both MSA and CA. The corpus planning and compiling phase covers the following steps: text normalization, automatic tokenization and text annotation.

Feasible initial tagging models design and implementation phase: The main objective of this phase is to find feasible baseline statistical Arabic POS tagging models. In other word, we want to know which is the appropriate size of context, one previous tag (bigram) or two previous tags (trigram), should be utilized to handle the Arabic POS ambiguity. In this phase, we have explored both bigram and trigram HMM graphical models to acquire and represent the language model. We also have investigated several sophisticated smoothing algorithms. The main objective of employing these smoothing techniques is to handle data sparseness problem and to study their influence on the overall tagging results. We also developed Arabic version of the (TnT) Trigram POS tagger which has widely been evaluated in several languages with notable success (Brants 2000; Raul & Alexander 2003). The output of this phase is six Arabic POS tagging models; each one is an independent POS tagger.

Tagging models improvement phases: The main purpose of this phase is to design and implement efficient and robust statistical Arabic POS tagging models. This can be achieved by enhancing and optimizing the POS tagging models from the previous phase. This phase takes as input the best tagging model from the previous phase.

However, this phase consists of three parts. The first part is about constructing efficient and robust statistical Arabic typical POS tagging models. To do so, we have proposed several new lexical models for unknown words. The second part is about the selection of the optimal segmentation levels for Arabic POS tagging. In this part, we also evaluate the influence of each segmentation level on the tagging performance of some of the tagging models that are produced in the first part. In the third part, we describe several methods for constructing a fully supervised stochastic for the morphosyntactic disambiguation problem. The methodology employed is based on the problem decomposition and the combination of several n-attributes morpheme-based probabilistic classifiers. The main output of this phase are around nine efficient and robust tagging models for both typical and fine-grained Arabic POS tagging.

Evaluation phase: The methodology used for the evaluation in this work follows EAGLES standard (EAGLES 1996; Giesbrecht 2008). In this phase, we evaluate the impact of different POS tagging approaches on the tagging accuracy, evaluate the impact of using different tag sets on the same texts on the tagging accuracy and evaluate the impact of using texts from different text types in training and testing on the tagging accuracy.

The experimental data, which is used to measure the performance of the proposed tagging models, are the Quranic corpus developed at Leads University(Dukes et al. 2010; Dukes & Habash 2010) and our corpus developed in the corpus planning and compiling phase.

1.7 THESIS OVERVIEW

Rest of this thesis is organized into chapters as follows:

CHAPTER II: Provides a review of the tagging techniques and of the related Arabic work. In addition, this chapter discusses the current research directions.

CHAPTER III: Provides a basic background for the Arabic language and its main characteristics. It also discusses the Arabic word derivation process, the existing Arabic POS tag sets and introduces those that have been used in this work. Finally , this chapter presents the resources used in this work: Arabic corpora and morphological analyzer tools

CHAPTER IV: Describes the research methodology employed by this research. In addition, the chapter provides the motivations of using HMM and the directions that followed to come with new efficient tagging models for Arabic. Finally, this chapter describes the methodology used to evaluate these tagging models.

CHAPTER V: Describes HMM based stochastic algorithms for Arabic POS tagging. In this chapter, different configurations of HMM models and smoothing algorithms have been investigated and

evaluated

CHAPTER VI: Investigates the characteristics of unknown words in Arabic and presents new methods to handle such problem. In addition, it evaluates the influence of the tag set granularity and the segmentation level on the tagging accuracy when only small amount of training data are avilable.

CHAPTER VII: Presents several methods to statistical morphological disambiguation for rich morphological languages based on the combination of several stochastic N-attributes classifiers.

CHAPTER VIII: Provides general conclusion, summarizes the thesis, presents contributions, and outline several directions for future work.

CHAPTER II

LITERATURE REVIEW

2.1 INTRODUCTION

Research on POS tagging has a long history. Numerous approaches have been successfully applied to POS tagging. These approaches can be classified into two main groups according to the nature of knowledge they use (Halteren 1999; Mārquez 1999; Màrquez et al. 2000): linguistic and machine learning family. Some languages have a rich and complex morphology, requiring designing advanced techniques that take into account large tag sets and high ambiguity rate.

The chapter is intended to be a broad survey of the state of the art in the main areas related with the thesis contents. This chapter provides a review of the existing POS tagging techniques. This chapter also provides a brief discussion on fine-grained POS (morpho-syntactic) disambiguation efforts in morphologically rich languages including Arabic. Further, we provide a description of the work on Arabic Language POS tagging. Finally, we also briefly discuss the current research directions.

2.2 LINGUISTIC TAGGERS

Research in automated POS tagging was firstly started in middle sixties and seventies (Harris 1962; Klein & Simmons 1963; Greene & Rubin 1971). Researchers manually designed rules for tagging. In fact, the earliest disambiguation algorithms were based on a two-phase architecture. In the first phase, each word is assigned its possible POS from a dictionary. While in the second stage, a large set of disambiguation rules are used to select only a single part-of-speech for each word (Dandapat 2009). The rule-

based taggers use contextual information to bring down the possible POS tags or to assign a POS tag to the unknown based on a set of language rules, which are often known as context frame rules. Since that time to nowadays, a lot of research has been devoted to improving the quality of the tagging process with respect to both accuracy and efficiency.

Recent rule-based taggers still incorporate the knowledge as a set of rules, or constraints, written by linguists. However, the current models are expressive and accurate and they are used in very efficient disambiguation algorithms. The linguistic rules range from a few hundred to several thousands, and they usually require years of labour (Thede 1999; Dandapat 2009).

The following will discuss in brief some of the well-known rule-based tagging systems:

- **CGC: Computational Grammar Coder system:** Klein and Simmons (1963) developed, as a part of their larger question-answering system, a Computational Grammar Coder (CGC) which itself a POS tagger. Their tagger utilizes several smaller English lexicons with 20,000 words, such as function-word dictionary. This dictionary comprises approximately 400 items of frequently occurring words, which are unambiguous, have unique grammar codes (tags) articles. It includes prepositions, pronouns, conjunctions, auxiliary verbs, adverbs, etc. Their CGC program tags words using these lexicons and the suffix information and the context frame. Furthermore, content words, about 1,500, that are exceptions to the computational rules are stored in a dictionary. They ran an experiment on several pages of the Golden book encyclopedia and reported that their system correctly tagged 90% of the words.

- **TAGGIT system:** Greene and Rubin (1971) developed the first pioneering tagger system for English , which is known as, TAGGIT. This tagger was used for initial tagging of the Brown Corpus. It was the first tagger which introduced the idea of providing a text corpus annotated with POS information as a useful tool for linguistic research. The TAGGIT used small lexicon containing about 3000 words, each word in lexicon was assigned its tag(s) manually. They used several

thousand context-frame rules to disambiguate those words have more than one tag. First, each word is checked to see if it is found in the exception lexicon, which contains all-known closed words. If the word is found on the lexicon and has one tag, this tag is extracted and assigned to the word. If the word is not found on the lexicon, then the word's ending is checked against a suffix list of about 450 suffixes. If it has more than one tag or it is not assigned any tags in the previous steps, a disambiguation routine, which contains a set of context frame rules, has been applied to assign the best tag to the word. The TAGGIT used a tag set of about 77 tags. When evaluated on Brown corpus, the accuracy of TAGGIT was 77 %. The rest was completed manually over a period of several years.

- **ENGCG: ENGlish Constraint Grammar system:** Voutilainen and Heikkila (1994), Karlsson et al. (1995) and Voutilainen (1995) implemented a constraint-based disambiguator system for English called ENGCG (ENGlish Constraint Grammar) for POS ambiguity and syntactic parsing. A set of 139 tags was used. ENGCG tagger consists of two main rule modules. The first one is a grammar specifically developed for POS ambiguities resolution while the one is a syntactic grammar for syntactic parsing which also solve the pending POS ambiguities as a side effect. Its grammar consists of 1,100 constraints. The first module disambiguated 93-97% of all the word. After the first module is applied, at least 99.7% of all words retained the contextually most appropriate POS tag with 1.04 POS tag per word on the average, and with an optionally applicable heuristic grammar of 200 constraints resolves about half of the remaining ambiguities 96-97% reliably. The ENGCG system was tested against a test corpus of 38,000 words and he reported it correctly tagged more than 98% of the words. Later, the ENGCG was combined with Xerox Tagger. In a 27,000 word unseen text, they reached an accuracy of about 98.5%, with no ambiguous word.

The Constraint Grammar formalism has also been applied to other languages, as Turkish (Oflazer & Kuruöz 1994) and Irish (Dhonnchadha 2008). The advantages of rule-based taggers are that rules can be hand-written and easy comprehended. Furthermore, a rule-based tagger can achieve good results. The main disadvantages of the rule based approach (Thede 1999; Reichel 2005; Spoustova et al. 2007):

1. Rules are language and tag set specific
2. Time consuming rule adjustment; takes a large amount of work
3. Needs a lot of linguistic knowledge
4. Lack of the generalization capability and missing the transferability to other languages.

2.3 MACHINE LEARNING AND POS TAGGING

The POS disambiguation also can be viewed as a classification problem: the tag set are the classes and an automatic classification method is used to assign each occurrence of a word to one class based on the evidence from the context. One of the important steps in POS disambiguation is the selection of the classification method. Most of the new approaches stem from the field of machine learning (Navigli 2009). These methods range from methods with full supervision, to fully unsupervised methods.

2.4 SUPERVISED APPROACHES: REVIEW

In the last 20 years, the NLP community has witnessed a significant shift from the use of manually crafted systems to the employment of machine learning classification methods (Navigli 2009). This increase of interest toward machine learning techniques is reflected by the number of supervised approaches applied to the problem of POS disambiguation. Given a sufficient amount of manually tagged documents, these approaches have demonstrated the ability to learn the instance of a tagging mechanism from manually tagged data and apply it successfully to unseen data. The training set used to learn the classifier typically contains a set of examples in which a given target word is manually tagged with a POS from predefined tag set (POS classes).Supervised approaches proved to be more accurate than other approaches. In the next subsections, we briefly review the supervised machine learning approaches that have been proposed for the POS tagging.

2.4.1 Transformation based learning

Brill (1992; 1993a; 1993b; 1994; 1995a) presented an innovative learning algorithm Transformation Based approach (TBL). It was inspired from both rule based and stochastic taggers. According to Brill (1992) in rule-based approaches, it is difficult to construct rules and in probabilistic approaches, much space is required to store the table of frequencies. TBL approach overcomes these issues by providing an automatic extraction of rules. Figure 2.1 which is reproduced from the original figure (Brill 1993b), illustrates the learning process of TBL tagger.

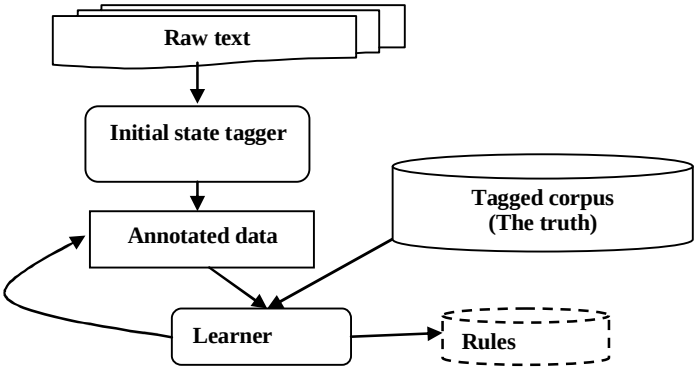

FIGURE 2.1 Supervised Transformation Based Learning
Source: Brill 1993b

First, unannotated text is passed through the initial-state annotator. This can range in complexity from assigning random tag to assigning the output of sophisticated manually created annotator such as n-gram tagger. In this step, the tagger assigns to every word its most probable POS tag, as estimated from the small annotated training corpus. The training set is used here only to determine the most likely (frequent) tag for each word. For unknown words, the most probable tag was guessed based on suffix and prefixes analysis up to 4 characters, previous and next word information and the appearance of special character in the word. For example, xxxxxxxion (where x represent any letter) would be tagged as a noun because this is (most likely) the most common tag for words ending in "ion. Secondly, the results are compared to the truth (manually tagged corpus). Transformation rules

can then be learned, which can be applied to the automatic annotated text to make it better resemble the truth.

A set of transformation templates is pre-specified. These transformation templates specified the types of transformations, which can be applied to the corpus. In all of the learning modules, the transformation templates are very simple, and do not contain any deep linguistic knowledge. The number of transformation templates is also small. The templates are of the form:

Change a tag from X to Y, if the previous tag is Z.

X, Y and Z are uninitialized variables. All possible instantiations of all templates define the set of allowable transformations.

TBL used both lexical and contextual information. The templates that used lexical information are as follows:

1. Change the most likely tag to X if:
2. Deleting (adding) the prefix (suffix) x, |x|< 5 results in a word.
3. The first (last) 1, 2, 3 or 4 characters of the word are x.
4. Adding the character string x as a prefix (suffix) results in a word (|x|< 5)
5. Word Y ever appears immediately to the left (right) of the word.
6. Character Z appears in the word.

From these lexical transformation templates, several transformations rules can be learned. For example:

Change any tag to *Present Part. Verb*, if the suffix is *"ing"*.

After applying the lexical transformation rules, the next step is to use contextual information to disambiguate words. The templates that used contextual information are as follows:

Change a tag from X to Y if:

1. The previous (following) word is tagged as Z.
2. The previous word is tagged as Z, and the following as W.
3. The following (preceding) 2 words are tagged as Z.
4. One of the 2 (3) proceeding (following) words is tagged as Z.
5. The word, two words before (after) is tagged as Z.

An example of a learned transformation is:
Change the tag from *VERB* to *NOUN* if the tag of the previous word is a *DET.*

TBL provides the ease of adjusting to new languages such as: i) templates of the rules can be defined based on sentence structure and characteristics of the languages, ii) a small set of human readable rules provide the ease of finding problems that affect the tagging accuracy and implementing the improvement (Chenda & Kameyama 2007a; Chenda & Kameyama 2007b). Chenda and Kameyama (2007b) used a combination model of TBL and trigram models. The trigram model chooses the most likely tag among the tags provided by rule guesser, and predicts any words if their internal structures don't match to any feature rules. This means that the unknown word is passed to the rule-based guesser. If the rule guesser cannot provide any tag for the unknown word, the word is passed to the trigram guesser.

2.4.2 Hidden Markov Model

Hidden Markov Model (HMM) is a well-known probabilistic model, which can predict the tag of the current word given the tags of one previous word (bi-gram) or two previous words (trigram). The HMM tagger assign a probability value to each pair $<w_1^n, t_1^n>$, where $w_1^n = w_1 \ldots w_n$ is the input sentence and $t_1^n = t_1 \ldots t_n$ is the POS tag sequence. In HMM, the POS problem can be defined as the finding the best tag sequence t_1^n given the word sequence w_1^n. This is can be formally expressed as:

$$t_1^n = \arg\max_{t_1^n} \prod_{i=1}^{n} p\left(t_i \mid t_{i-1} \ldots t_1\right).p\left(w_i \mid t_i \ldots t_1\right) \qquad 2.1$$

This is actually impossible to calculate because of the sparse problem, especially, if n is big. In practice, bi-gram or tri-gram model is often used to alleviate data sparseness. The probabilities are estimated with relative frequencies from the training data.

One of the first taggers based on Markov model was by Garside (1987) and Marshall (1987) and targeted to tag Lancaster - Oslo/Bergen (LOB) corpus (Johansson et al. 1986). This tagger was based on the use of probabilities of bigram tag sequences, with the limited use of the higher context. However, the probability of having a different part of speech for a particular word was assigned by heuristic method. (Church 1988) also used second order Markov model. They trained their systems on large handed tagged corpora. Using this tagger, they are able to tag accurately more than 96% of the test set. Thede (1999) and Thede and Harper (1999) used full second order Markov model. They used trigram model not only for the contextual probability but also for the lexical and suffix probability. Their tagger accuracy outperforms standard trigram tagger, memory based tagger and maximum entropy tagger for the overall accuracy in both closed and opened vocabulary assumptions.

Trigrams'nTags (TnT) developed by Brant (2000) is a statistical approach, based on a second order hidden Markov model. TnT uses the Viterbi algorithm with beam search for decoding. The states represent tags, and the transition probabilities depend on pairs of tags. The system uses the relative frequencies. The main smoothing technique implemented is the linear interpolation of unigram, bigram and trigram probabilities. The system uses a context of three tags. Unknown words are handled by suffix analysis up to the last ten characters of the word. Additionally, information about capitalization is included as default. Anwar et al. (2007) showed that overall tagging accuracy of HMM tagger could be improved from 90% to 96% using different smoothing methods.

2.4.3 Maximum Entropy Model

The Maximum Entropy (ME) framework estimates probabilities based on the principle of making as few assumptions as possible, other than the constraints imposed. Such constraints are derived from the training data, expressing some relationships between features and outcome Maximum (Berger et al. 1996; Chieu & Ng 2002). Ratnaparkhi (1996; 1998) developed the Maximum Entropy Part Of Speech Tagger (MXPOST). It is a probabilistic approach based on a maximum entropy model

where contextual information is represented as binary features that are used in order to predict the POS tags. The binary features used by MXPOST include the current word, the next and previous two words and the preceding one or two tags. For rare and unknown words, the first and last four characters are included in the features, as well as information about whether the word contains uppercase characters, hyphens or numbers. The tagger uses a beam search in order to find the most probable sequence of tags. The tag sequence with the highest probability is chosen. The overall accuracy is 96.3 %for known words and 85.56% for unknown words. Toutanova and Manning (2000) improved the accuracy that is achieved by Ratnaparkhi (Ratnaparkhi 1996). They analyze the errors and increase the features to include features for disambiguating proper noun, features for disambiguating tense forms of verbs and features for disambiguating particles from prepositions and adverbs. Zhao et al.(2007) use Bound Limited Memory Variable Metric Method (BLMVM) to estimate MXPOST parameters.

2.4.4 Conditional Random Field

Conditional random fields (CRF) are a probabilistic framework for labeling and segmenting structured data, such as sequences, trees and lattices. The main advantage of CRF over HMMs is their conditional nature, resulting in the relaxation of the independence assumptions required by HMM in order to ensure tractable inference. Additionally, CRF avoid the label bias problem, a weakness exhibited by maximum entropy Markov models (Lafferty et al. 2001; Wallach 2004).

CRF has been introduced by Lafferty et al. (2001) for segmentation and sequence labeling tasks. They also used it for POS tagging and compared the accuracies of several supervised POS tagging models, while examining the effect of directionality in graphical models. By adding a small set of orthographic features, the overall error rate reduced by around 25% and the out-of-vocabulary error rate reduced by around 50%.

2.4.5 Other Supervised Methods

Support Vector Machine (SVM) is a supervised machine-learning algorithm for binary classification and for multi-class classification. In this method, data consisting of two categories is classified by dividing space with a hyperplane. In basic form, a SVM learns to find a linear hyperplane that separate both positive and negative examples with maximal margin (Witten et al. 2005). Nakagawa et al.(2001) and Nakagawa (2006) describe the POS prediction for unknown words using Support Vector Machines. They achieved high accuracy in POS tag prediction using substrings, surrounding context and tags of surrounding context as the features. They integrate this method with a practical English POS tagger. SVM has good properties and performance, but their computational cost is large (Nakagawa et al. 2001). Beside the features used in Nakagawa et al. (2001), Giménez and Màrquez (2004) use other features such as sentence information, ambiguity classes and word length. Their SVMTool is trained for both English language using Penn Treebank corpus and Spanish language. SVM also used for POS tagging for other languages such as Dutch (Poel et al. 2007), Bengali (Ekbal & Bandyopadhyay 2008) and Chinese (Zhang et al. 2009; Wang et al. 2010).

Cyclic Dependency Network (CDN) is a supervised conditional Markov Model POS tagging which exploited information coming from both left and right contexts i.e. it uses the previous words tags and the following words tags. It is unlike all other graphical models like HMM, MEM and CRF that use only tags of the previous words. Toutanova et al. (2003) proposed this method to overcome the short dependency of HMM. They develop English tagger using CDN. When using tag to the left and tag to the right as features in addition to the current tag, accuracy improved to 97.24%. Tsuruoka and Tsujii (2005) present another way of making use of future tags i.e. the tags of words in the right side of the current word. In their inference method, they consider all possible ways of decomposition and choose the "best" decomposition. Accuracy on the Penn Treebank using full directional inference and unidirectional inference are reported. Full directional inference achieved higher accuracy than unidirectional ones.

Memory Based Learning (MBL) is a form of supervised, inductive learning from examples. During the training, a set of examples are stored in the memory. Each case consists of a word (or a lexical representation for the word) with preceding and following context and the corresponding category for that word in that context. The tag of the word in given context is predicted using similarity reasoning (Daelemans & Bosch 2009). Memory-based learning is an expensive algorithm. Of each test item, all feature values must be compared to the corresponding feature values of all training items (Daelemans et al. 1996). Daelemans et al. (1996) proposed MBL tagger for English language. They used IGTree algorithm to achieve good performance and to reduce the time complexity of the memory based learning. The memory is reformulated as a tree so it becomes easier to get the similar case of the current context with a reasonable time. To handle the unknown they use both morphological features and the contextual features.

Decision Tree Model: The main problem of the probabilistic models is the sparse data problem mainly, when they are trained on small data. A decision tree is a predictive model used to represent classification rules with a tree structure that recursively partitions the training data set. Each internal node of a decision tree represents a test on a feature value, and each branch represents an outcome of the test. A prediction is made when a leaf node is reached. Schmid (1994b) present a decision tree to pass up the sparse data problem and to obtain reliable estimates of the transition probabilities. The decision tree has the ability to determine the appropriate size of context to estimate these probabilities. A suffix lexicon is also used to tag the unknown word. Màrquez and Rodríguez (1998) and Màrquez (1999) develop a decision tree tagger for English POS tagging. They use non-incremental supervised learning from examples of TDIDT (Top Down Induction of Decision Tree) to construct the decision tree. They handle unknown words tagging, by first assuming that unknown words belong to the opened class categories, and then they use the word form information and the context information.

Artificial Neural Network: a neural network is an interconnected group of artificial neurons that uses a computational model for processing data based on a connectionist approach. Pairs of input feature and desired response are input to the

learning program. The aim is to use the input features to partition the training contexts into non-overlapping sets corresponding to the desired response. Input layer consists of a set of units equal to the number of tags in the tag set. For each word, all tags with which a word was marked are activated. Network knows about the right tag due to the training and deactivates other output units. Schmid (1994a) present a neural network tagger (Net Tagger) with a context window of three preceding words and two succeeding words, which was trained on the Penn Treebank corpus. They use single perception in order to predict the correct POS tag of an input word. Ma et al.(1999) use three layer perceptions with elastic input. Ahmed et al.(2002) trained a multi-layer perception network tagger with error back–propagation-learning algorithm. The tagger is trained with corpus of size 156622 words. It is tested on data include unknown words. Recently, Poel et al.(2008) developed a neural network tagger for Dutch

Classifiers Combination: Combining classifiers is well-known technique in the NLP and Machine Learning community. The key ideas behind combining individual classifiers are that every individual classifier produces different type of errors, and also classifiers are combined to exploit their strengths. Ensemble methods are becoming more and more popular as they allow one to overcome the weaknesses of single supervised approaches (Navigli 2009). The combined tagger always outperforms the best of its individuals (Brill & Wu 1998; Halteren et al. 2001; De Pauw et al. 2006; Kuta et al. 2008; Kuta et al. 2010). The classifiers combination techniques works as follow: first every classifier (tagger) produce the tag of the current word; then, a selection algorithm selects the right tag among all the N results of the N taggers. The selection algorithm is the heart of this methodology. The selection algorithm determines the accuracy of the combined classifiers by choosing the best answer among a set of N answers. Some of the selection algorithms are based on majority, plural voting, tag precision and stacking (cascade classifiers).

2.4.6 Supervised Machine Learning Techniques Analysis

In general, most of the supervised machines learning techniques for POS tagging have accuracy between 95-97%, when they are trained and tested with data from the same domain. However, the accuracy of a machine learning tagger is affected by the