Combining corpus-based and linguistic models for Arabic speech systems

Hanady Ahmed

Arabic Department, CAS
Qatar University

hanadyma@qu.edu.qa

Allan Ramsay

School of Computer Science
University of Manchester

Allan.Ramsay@manchester.ac.uk

A truth:

- “Computers can do a lot of things but computers are not good at thinking about themselves. They really need to be spoon-fed the details”(Hetland.M, 2003).

The project

- This project is a joint project with Manchester university .
- It has been funded by the internal grants schema of Qatar University 2010-2011.
- Qatar University and Manchester university have extended this project to be : “Arabic Speech Recognition and Understanding : A hiypred approach“, which is funded by QNRF in the third cycle of NPRP projects (2010-2013)

Which Arabic Speech Systems?!

- Automatic generation (text-to-speech synthesis (TTS)) and recognition of spoken Arabic speech (automatic speech recognition (ASR)) is a challenging task. (The current presentation will focus on NLP for TTS)

- Automatic generation and recognition of any language is hard enough, but Arabic has a number of properties that make it even harder.(We are still in the first stage for designing speech recognition system for Arabic)

Scope of the research

- The main aim of the proposed research, however, is to extend the natural language processing engine (NLP) –rule based- so that it can also be used as the basis for a language model for TTS and speech recognition.
- Speech recognition engines require a 'language model' to help constrain the search for words that match the acoustic properties of the speech signal. Such language models are typically supplied as context-free grammars.

Scope of the research (Cont.)

- The existing linguistic engine can be used to produce analyses of input text which can in turn be used to convert written text – to- speech signal and to generate a context-free grammar of the kind that is required for speech recognition.
- In order to use the current engine for these tasks, we need to add corpus-based information, e.g. statistical part-of-speech tagging, probabilities relating to various non-canonical word orders, converting grapheme-to allophone (GTA) rules, and to extend the lexicon.

The Challenges !!!

- In particular, the *non-concatenative* nature of Arabic morphology and the range of permitted *word orders* mean that is very hard to provide language models of the kind that are required for deriving speech synthesizers or for training speech recognizers.
- The lack of *diacritics* in written Modern Standard Arabic (MSA) make it difficult to determine the underlying phonetic forms required for speech synthesis.

E.X: ktb /katab/”wrote” , /kutub/ “books”, /kattab/ “made s. to write” , /kutib/ “been written”,…..

1- Word Morphological structure

- Arabic grammarians traditionally described all Arabic words into three main lexical categories: Verb, Noun, and Particle. These categories could be classified into further sub-classes which collectively cover the whole of the Arabic language.

- Morphologically, Arabic is very rich and based on root-pattern structure. Most Arabic words are generated out of a finite set of roots (about 7000) transformed into stems using one or more of patterns (about 125). In theory, a single Arabic root can generate hundreds of words (noun, verbs). Arabic words may exist in hundreds of shapes in normal text by adding certain suffixes and prefixes (Kiraz 2000; El-Affandi 2002). Most of those patterns are nominal patterns.

SurafceForm	k	**aa**	t		b
Root Tire	**k**	**#**	**t**	**#**	**b**
Vowel Tire		**aa**		***i/a***	
UnderlyingForm	k	#:**aa**	t	**#**	b
FullForm	***k***	**aa**	***t***	*i*	***b***

Figure (1): Multi-Levels of diacritization

2- Sentence Structure

- Free Word order: Arabic sentence structure allows free movements for arguments of sentences around the predicate, for example, Arabic allows six logically possible word orders for simple verbal sentence **VSO** (with definite subject).
- Nominal Sentences:A nominal sentence is one where the subject precedes the predicate (Mohammed 2000) . The subject and the predicate has joined together without a copula.
- Construct phrase:Arabic allows an NP to function as a construct phrase that has the semantic relations as the possessive meaning in English. The two nouns in Arabic are joined together without any overt marker as:

  - ktaab? aalmdrs+i 'teacher's book'.

  case marker? +gen

- Zero subject: Main argument in a verbal sentence is a subject which could be deleted ,i.e, or has value zero as we have treated it.
- - katab aaldars+a 'he wrot the lesson'

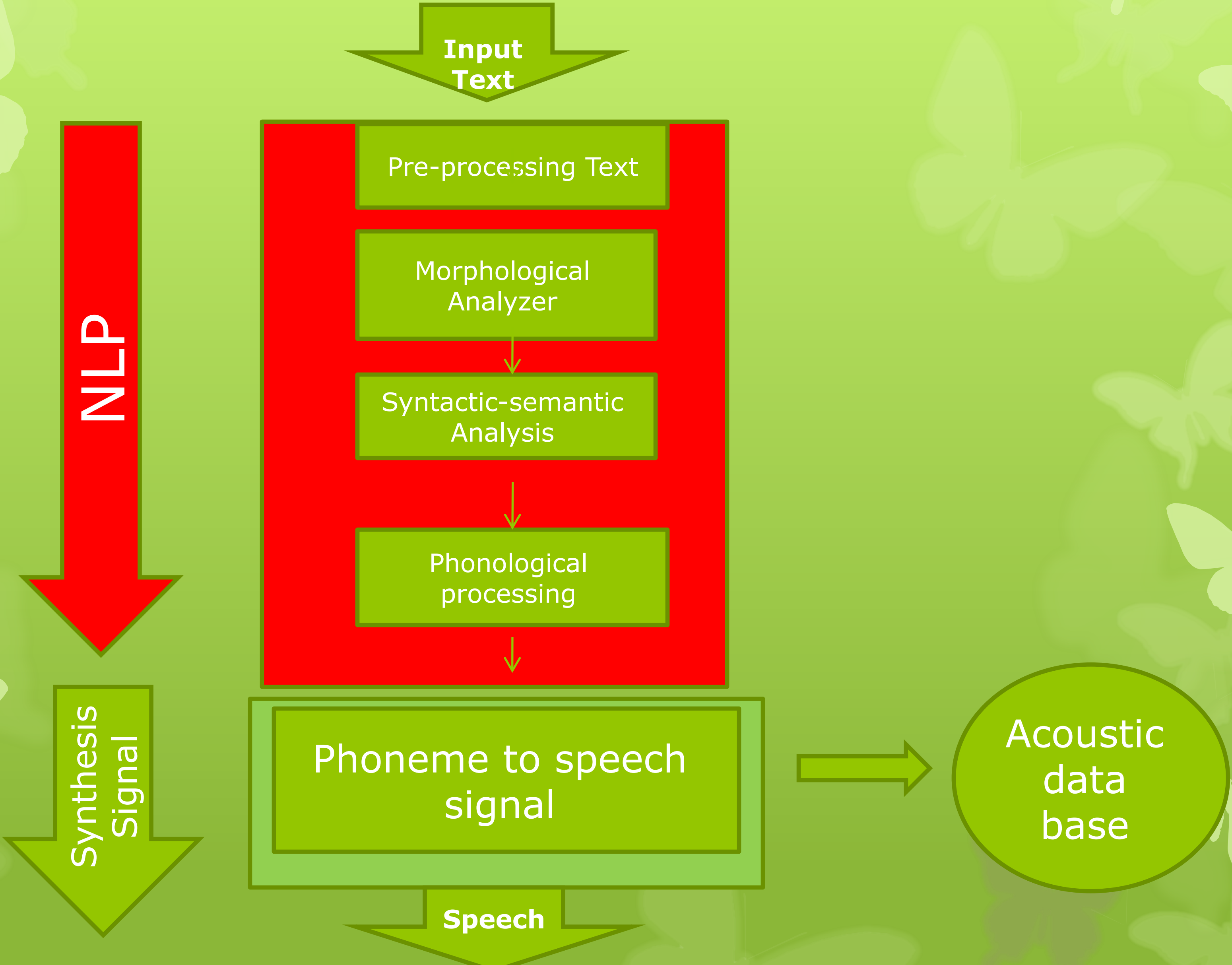
  V zero subject Obj

NLP Engine for Arabic TTS: Rule-based

- We have aimed to provide a text-to-speech system for modern standard Arabic (MSA) that has concentrated on handling the next issues:

- Diacritic assignment: (i.e. of recovering phonetically relevant information, such as choice of short vowels, which is not explicitly provided in the surface form of MSA). This is clearly a crucial issue: you can hardly produce intelligible spoken output if you do not know what the vowels are.
- Converting GTP : We describe an approach to the task of generating phonetic transcription from MSA text .
- Intonation Contour : The Engine also provides the information required for imposing an appropriate intonation contour for the Arabic sentences.

Linguistic Model: Text to Speech System (TTS)



Diacriticisation Mechanism

- We follow fairly standard practice by describing a word in terms of a template and a set of fillers (e.g. (McCarthy and Prince, 1990)).
- We use a categorial description of the way roots and affixes combine (Bauer, 1983); in order to improve the efficiency of the process of lexical lookup.
- We store the lexicon as a lexical tire and FST.
- We add a set of spelling rules to account for the variations in surface forms that are observed under various conditions.(details will be explained for Weak verbs)

Computational framework

- {struct(positions(start(0), end(1), span(1), +compact, xstart(0), xend(1)),

```
forms({y,a,k#t#b,0,uuna}, yktbwn))),
morph(diacrits(choices(actvPres(["0", "u"]),actvPast(["a", "a"]),
                       psvPast(["u", "i"]),psvPres(["0", "a"])),
                       actual(["0", "u“]))),
lextype(regular(i(1, "u"), a, 1))),
    syn(nonfoot(head(cat(xbar(+v, -n)),
         agree(third(+plural)),
         gender(-neuter, +masculine, -feminine)),
         vform(vfeatures(finite(+tensed, -participle, -infinitive),
                        -aux,
                        +active,
              view(tense(+present, -past, -future, -preterite, -free),
              subcat(args(["NOUN", "NOUN"]), fixed),
   foot(wh([]))),
   remarks(score(0))}
```

Computational framework (cont.)

- Input a sentence in arabic.

|: aaldrs

Found one

None like it. This one is no. 1

Everything we need should be encoded in the following list

[?,a,l,+,d,a,r,0,s,+,0,+,0,+,0,+,?,&]

This has now been changed into a list of phones

[phoneme(char(?), -vowel),
 phoneme(char(a), +vowel, -long, boundary(+morpheme)),
 phoneme(char(d), -vowel),
 phoneme(char(d), -vowel),
 phoneme(char(a), +vowel, -long),
 phoneme(char(r), -vowel),
 phoneme(char(s), -vowel)]

- Input a sentence in arabic
- **|: ‘lm aalTalb.**

Pitch markers have now been added

[phoneme(char(\`),-vowel),

phoneme(char(a),+vowel),

phoneme(char(l),-vowel), /

phoneme(char(l),-vowel),

phoneme(char(a),+vowel,-long,

pitch(pmark(high), FA),

stress(stressed)), /

phoneme(char(m),-vowel,boundary**(+morpheme**)),

phoneme(char(a),+vowel,

-long,

boundary(+morpheme, **+word)),&***

phoneme(char(?),-vowel,+**emphatic**),

phoneme(char(a), +vowel,-long,boundary(+morpheme),+emphatic),

phoneme(char(T),-vowel, +emphatic),

/

phoneme(char(T),+**emphatic),**

phoneme(char(a),+vowel,+long,+emphatic,

pitch(pmark(high), FB),

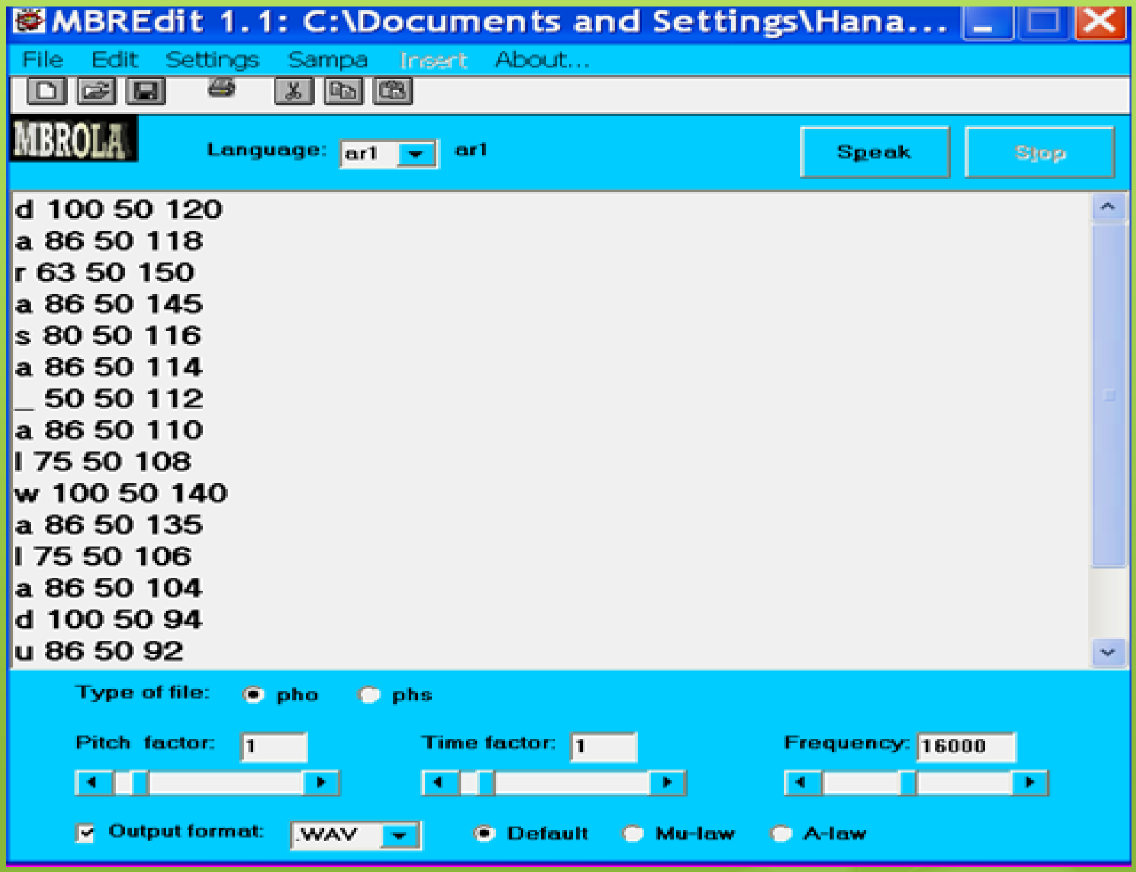
stress(stressed),

NLP output

- | ?- in arabic.
- Input a sentence in arabic

|: drs aalwld.

| ?- retrieve(19,P), syllabify(P,Q).cspeak('sound.pho', Q).



The Existing Linguistic Models

- The analyses produced by the linguistic engine are fine-grained dependency trees, annotated with a variety of syntactic and Morphological features.

- The linguistics models provides a phonological analysis for Arabic words and sentences ,i.e, converting written form into narrow phonetic transcriptions with assigning stress and generating intonation contour.

Limitations

- Small Lexicon contains hundred of entries.
- Processing marked and un-marked short simple sentence.
- Small ontology for sentences disambiguation.
- The main aim of the corpus-based NLP engine is to improve the performance of the existing engine in the face of long sentences and a wide vocabulary, by adding statistical evidence to the existing rule-based approach and by extending the lexicon using resources such as Pen Arabic Treebank , Buckwalter Arabic morphological analyzer.

Corpus

- Backwater Morphological Analyzer:
- ➢ DictStems:

sense: FullForm:HisAb_1//Translation:calculation

[[(SurafceFrom:'HsAb', FullForm:'HisAb', Tag:'N', 'calculation', '')]]

- ➢ Penn Arabic Treebank (PAT) : Treebank V.I.4.

Corpus-based NLP Engine

- We faced a number of challenges:
- Merging Lexicons: Automatically extracting the lexical entries from BW lexicon and converting to our System notations.
- TagSet: Understanding BW classifications for the Lexemes (Verbs and Nouns).
- Filling the missing information in BW dictStems.
- Reclassification of senses.
- Checking sense translations.

First Stage: Merging Lexicons

- Thus the first stage of the research involves exploring ways of getting better information out the BW lexicon to leverage a large fine-grained lexicon of the Existing system (PARASITE).

- We will see the details in the next set of the slides:

Lexicon: Nouns

- **BW. Entry:**

```
/*
k?t?b
sense: FullForm: kAtib_1//Translation:clerk TagSet:(N/ap)
[[(SurfaceForms'kAtb', FullForm:'kAtib', 'N/ap', 'clerk', '')]]
*/
```

- **Parasite Entry:**

```
"k?t?b" lextype regular(nominal,
            ['':[["A","i"]:_:regular(''):thing: masculine:[translation('clerk')]]], 1)
      ::: noun delayed ntype(simpleArabic).
```

Parasite output using BW lexicon: nominal Lexeme

- **| ?- in arabic. kAtb^**

| ?- | ?- underlyingForms.

3 -> {{{{{k?t?b,o(*deriv(1))},o(emptygender(*gender))},{_3887}},o(emptyDet)},{_3883}} (kAtib+?+?, clerk: masculine: no of args=0)

2 -> {{{k?t?b,o(*deriv(1))},o(*tense)},{a}} (kAtab+a, correspond with: no of args=2, +active)

- **| ?- in arabic. kAtbAn^**

| ?- underlyingForms.

2 ->{{{{{k?t?b,o(*deriv(1))},o(emptygender(*gender))},Ani},o(emptyDet)},{_3964}} (kAtib+Ani+?, writer: no of args=0)

- **| ?- in arabic. kAtbwn^**

%% justWords wasn't set%%

::: %%%% Parse completed -

Lexicon: Verbs

- **BW sense:**

```
/*
sense: Hasib-i_1//regard
[[('Hsb', 'Hasib', 'PV', 'regard', ''), ('Hsb', 'Hosib', 'IV', 'regard', '')]]
*/
```

- **Parasite Entry:**

```
"H?s?b" lextype regular([["a", "i"], ["o", "i"], ["a", "i"], ["o", "i"]], a, 1)
    ::: verb
    delayed vtype(valency(1, [agent:living, object])).
```

Parasite output using BW lexicon: verbal Lexeme

|: yes| ?- | ?- in arabic. yktb^

Input a sentence in arabic

/**** DEPENDENCY TREE ***************

{{{{yu},{l,k?t?b}},o(tns1)},{_20215}}

---------------------------------****/

- This analysis had the following problems: _11714+_11715|:
- yes| ?- | ?- underlyingForms.
- 2 -> {{{{yu},{l,k?t?b}},o(tns1)},{_3524}} (yulkotib?, dictate: no of args=2, +active)
- 3 -> {{{{ya},{k?t?b,o(*deriv(1))}},o(tns1)},{_3564}} (yakotub?, write: no of args=2, +active)
- 4 -> {{{{yu},{k?t?b,o(*deriv(1))}},o(tns1)},{_3747}} (yukat~ib?, make write: no of args=3, +active)
- 5 -> {{{{yu},{l,k?t?b}},o(tns1)},{_3396}} (yulkotib?, dictate: no of args=1, +active)
- 6 -> {{{{yu},{l,k?t?b}},o(tns1)},{_3322}} (yulkotab?, dictate: no of args=1, -active)
- 7 -> {{{{yu},{k?t?b,o(*deriv(1))}},o(tns1)},{_3541}} (**yukat~ab?**, make write: no of args=2, -active)
- 8 -> {{{{yu},{k?t?b,o(*deriv(1))}},o(tns1)},{_3358}} (yukotab?, write: no of args=1, -active)
- Yes

- | ?- Input a sentence in arabic

|: yktb Alrjl Aldrs

| ?- underlyingForms.

5 -> {{al,{{r?j?l,o(*deriv(1))},o(emptygender(*gender))}},{_3531}} (al+rajul+?, man: no of args=0)

6 -> {{al,{{{d?r?s,o(*deriv(1))},o(emptygender(*gender))},{_3928}}},{_3926}} (al+daros+?+?, lesson: no of args=0)

2 -> {{{{yu},{I,k?t?b}},o(tns1)},{_3552}} (yu+I+kotib+?, dictate: no of args=2, +active)

3 -> {{{{ya},{k?t?b,o(*deriv(1))}},o(tns1)},{_3590}} (ya+kotub+?, write: no of args=2, +active)

4 -> {{{{yu},{k?t?b,o(*deriv(1))}},o(tns1)},{_3708}} (yu+ka~tib+?, make write: no of args=3, +active)

Yes

|

Weak Verb

- Weak verbs are in fact regular verbs whose spelling reflects a small set of phonological contractions.

  e.x: "w#q#f, q#w#l, r#m#y"

- Our analysis allows us to obtain 'underlying forms' for the surface forms of weak verbs which show how they are related to their roots.
- Bw lexicon does not play a significant role for treatment Weak verbs. Therefore , we edited our weak verb conjugation tables and **spelling rules**.

Spelling rules

- **1- Character**:

```
character(char(و(w)),
          underlying("w"),
          vc(+vowel,+consonant, +long)).
```

- **2- Format:**

  /L/ P /R/=> Q (Chomsky and Hall 1968)

- **3- The rule:**

```
%% 't$kyAn'=['tu$okawAni']
 [y]
  ==>
  [{w, +final}] :
  [_, "a"] ## ['A', x0, _] : X:-
             language@X <> arabic,
             -affix@X.
```

System analysis:

- | ?- runTests('$kw').

/*3rd dual f*/

Sentence: 44

runGrammarTest('t$kyAn'=['tu+$okaw+Ani'], _).

107 ->

{{{{tu},{$?k?w,o(*deriv(1))}},o(tns1)},Ani} (tu+$okaw+Ani, unknown: no of args=1, -active)

Expected surface forms found: ['tu+$okaw+Ani']

Expected number of analyses found: 1

****

Tagger

- Version 1: trained on classical Arabic, where it achieves 95% accuracy over a set of about 15 tags.
- Version 2: trained on Penn treebank, 96.4% over 43 tags, 91% over 306 tags

E.X:

The tagset includes markers for various kinds of clitics, so that we classify ?akatbtuhum آكتبتهم؟ ,for instance, as qmarker+ V+PRO .

Parasr

- Initial experiments using trainable dependency parsers achieve around 80% accuracy: not good enough to be relied on (trained on 4000 sentences from Penn treebank, tested on 1000).
- But good enough to provide a guide to the rule-based parser, which is very slow on long sentences.

This is currently under development.

Conclusions

- The basic problems of Arabic morphology are well known. A single word may have numerous forms, marking various syntactic features.
- We present a treatment of Arabic morphology which covers the standard cases, but which has two significant advantages:
- ✓ (i) We delay making decisions about the underlying form until we have the information that is necessary for getting the decision right.
- ✓ (ii) We can take account of the phonological processes that produce the varying forms of ‘weak’ verbs without having to declare these verbs as belonging to a special class.

Evaluation

- Combining corpus-based and rule-based linguistic models provide:-

- A lexicon which has approximately 33,000 entries.
- A training data for test the efficiency of the tagger.
- A trainable dependency parsers to guide the rule-based parser and to achieve high accuracy.

Future Work

- Recently, we have got another two kind of corpus: SAMA analyzer and Prague Treebank.

Questions

Thank You