The experience of building industrial-strength parser for Arabic

Dictum Ltd, Nizhny Novgorod, Russia

Dmitriy Strebkov, Nadezhda Hilal, Abdellah Redjaimia, Dan Skatov

Background

Why we analyze Arabic?

What tools do we use?

- Dictum's **syntactic parser** with language-independent core that is based on hybrid approach;
- "key-value" model for storing linguistic information with efficient access procedure (**DAWG**);
- **Stanford Part-Of-Speech Tagger**

Parser: Linguistic part

Chronology of morphological analyzer choice:

Buckwalter Arabic Morphological Analyzer (BAMA)

ElixirFM (redesigned & extended BAMA lexicon, inspired by Functional Morphology library for Haskell)

DAWG+ElixirFM (storing linguistic information returned from ElixirFM in DAWG key-value structure)

Parser: Rule base part

Syntactic rules that allow to acquire dependency and constituency parses simultaneously:

```
//كأن المطر سيهطل - ليت الشباب يعود  "as if the rain come - wish
youth returned"
Action+Entity+SpecialFuncWord  {
  T: [Action] <> [Entity] <> [SpecialFuncWord]
  C: LI2.Case == CASE_ACC && PH1.Type != PHRASE_IMPERATIVE_ACTION
  && LI1.Gender == LI2.Gender;
  Main: 1; L: 1=>PredSubj=>2; 2=>Auxiliary=>3;
}
```

- T (Template);
- C (Criterion);
- M (Main Phrase);
- L (Link(s) for Dependency Tree);
- A (Action)

Parser: The structure of the algorithm

Modified Cocke — Younger — Kasami (CYK) algorithm

RTL-specific adaptation of the algorithm: phrases are added to the CYK matrix in reverse order:

اللاعبون المخلصون

^{2}المخلصون : Adj_1 ^{3}المخلصون : $Noun_2$	**$Noun_3$: Rule=Adj+Entity** $Noun_4$: Rule=Entity+Noun
	^{1}اللاعبون : $Noun_1$

Parser: Range & Cut procedure

Ranging procedure is used to overcome the effect of high ambiguity in Arabic by ranging phrases in cells.

Ranging parameters that are currently used:

1. POS tagger approvals count;
2. Inverted links count;
3. Weight.

Make a **cutting** after ranging to leave only K-best phrases in each cell.