

Simultaneous Morphological Analysis and Lemmatization of Arabic Text

Rushin Shah

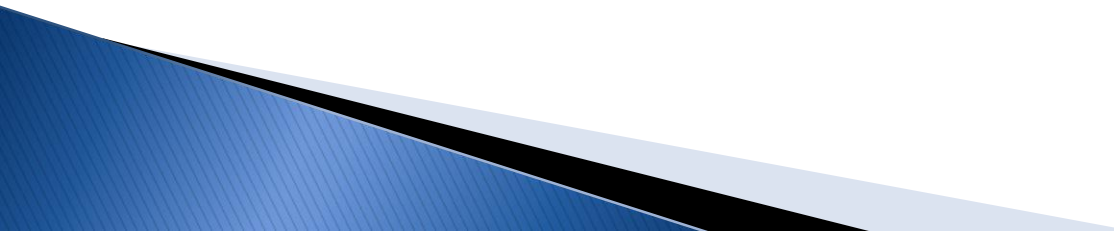
Linguistic Data Consortium

Under the guidance of Prof. Mark Liberman, Prof. Lyle Ungar and Mr.
Mohamed Maamouri

Motivation

- ▶ Arabic corpus annotation currently uses the Standard Arabic Morphological Analyzer (SAMA)
- ▶ SAMA generates various morphological and lemma choices for each token; manual annotators then pick the correct choice out of these.
- ▶ The problem is, there are dozens of choices for each token – typically 20 to 80
- ▶ No information about which choice is more likely
- ▶ Considerable time expended per token
- ▶ Would be nice to cut down the no. of choices to 1 or 2

Motivation

- ▶ Current Arabic language helper tools such as Al Kitaab provide labels for specific texts only, and hence in a sense, are offline.
 - ▶ No interface for user to supply text and obtain labeled version of it back
 - ▶ Intermediate learners would find helpful a system that allowed them to submit news articles, and supplied them annotated versions of those.
- 

Challenges

- ▶ Arabic has complex morphology, especially inflectional
- ▶ Large no. of morphosyntactic features (MSFs) such as basic POS, gender, number, mood, case, etc.
- ▶ Agglutinative: clitics (prefixes, suffixes), attached pronouns
- ▶ Thousands of morphological labels, compared to only about 50 for English
- ▶ Diacritization: diacritics omitted in written text, so same token often has more than one meaning

Challenges

- ▶ Another problem arises due to the goal of simultaneity
- ▶ Lemmatization is similar to word-sense disambiguation, requires local context
- ▶ For example, if token t is in document d amongst set of documents \mathbf{D} , d is more useful in predicting the word-sense of t than \mathbf{D}
- ▶ However, for morphological analysis, global context is more useful.
- ▶ We need an approach that effectively uses both local and global context

Problem

- ▶ We take a corpus C of labeled Arabic text. Here, each label l is a combination of a lemma and a morphological analysis. This can be expressed as:

$$l = (\text{lemma}, \text{morphology})$$

- ▶ We also use a morphological analyzer, which returns for each Arabic token t a set of possible label choices $L(t)$. Let l denote such a possible label choice.

Problem

- ▶ Given this information, we wish to learn a function f such that:

$P(l) = f(t, l, C)$, where P is the probability of l being the correct label choice for token t .

Problem

We use the following two forms of corpora:

- ▶ A corpus C_{global} which serves as the global context. From it, we will extract features that are relevant for determining morphology.
- ▶ A corpus which we shall use to obtain instances for training and testing our model; this is our local context C_{local} .

Problem

- ▶ Let C_{train} and C_{test} denote the portions of C_{local} used for training and testing respectively.
- ▶ It is clear that $C_{\text{local}} = C_{\text{train}} + C_{\text{test}}$
- ▶ Also, we use the symbol C from now on to denote the union of C_{local} and C_{global} corpora
- ▶ Hence, $C = C_{\text{local}} + C_{\text{global}}$

Problem

- ▶ The function f can now be expressed as:

$$P(l) = f(t, l, C_{\text{local}}, C_{\text{global}})$$

- ▶ We wish to learn such a function f and ensure that f is as accurate as possible. In other words, if l' is the choice assigned the highest probability by f , f should maximize the probability $P(l' = l_{\text{answer}})$.

Resources

- ▶ We use a set of three modules of the Penn Arabic Treebank (ATB), namely ATB1, ATB2 and ATB3 as our corpus C of labeled Arabic text.
- ▶ Each ATB module is a collection of newswire data from a particular agency.
 - ATB1 uses the AFP as a source
 - ATB2 uses Ummah
 - ATB3 uses Annahar.

Resources

- ▶ We use the bulk of each module of ATB as our global context C_{global}
- ▶ The remainders of these modules form our local context C_{local}
- ▶ We divide C_{local} into 10 disjoint subsets of equal size, and perform training and testing using 10-fold cross validation.
- ▶ C_{train} and C_{test} are formed in a 7:3 ratio.

Resources

- ▶ We also use the Standard Arabic Morphological Analyzer (SAMA) as our morphological analyzer to obtain label choices for each token.
- ▶ For each token t , $M(t)$ denotes the set of label choices returned by SAMA for t .
- ▶ A sample output of SAMA is shown in the following table.

Resources – SAMA Output

Token	Lemma	Vocalization	Segmentation	Morphology	Gloss
yHlm	Halam-u_1	yaHolumu	ya + Holum + u	IV3MS + IV + IVSUFF_MOO D:I	he/it + dream + [ind.]
yHlm	Halam-u_1	yaHoluma	ya + Holum + a	IV3MS + IV + IVSUFF_MOO D:S	he/it + dream + [sub.]
yHlm	Halam-u_1	yaHolumo	ya + Holum + o	IV3MS + IV + IVSUFF_MOO D:J	he/it + be gentle + [jus.]
wbAltAly	tAliy_1	wabiAlt~Aliy	wa + bi + Al + tAliy	PART + PREP + DET + ADJ	[part.] + with/by + the + following/s ubsequent
AlSfHp	SafoHap_1	AlS~afoHap	Al + SafoH + ap	DET + NOUN + NSUFF_FEM_ SG	the + page/leaf + [fem.sg.]

Instances - Training

- ▶ We supply instances to the model of the form (t, l, C) where t is a token, l is a label choice, and C is the context of t .
- ▶ During training, an answer value of 1 or 0 is supplied with each instance depending on whether l is the correct label choice for t or not, respectively.
- ▶ For a token t in context C , in addition to an instance (t, l, C) where l is the correct label choice, there must also be instances where l is not the correct choice, else the model will train only on positive instances.

Label Choices - Training

- ▶ During training, for each token t we construct the set $L(t)$ of label choices as follows:
- ▶ We add to $L(t)$ all the label choices contained in $M(t)$.
- ▶ We also add to $L(t)$ the correct label l_{answer} for token t if it isn't already included in $M(t)$.
- ▶ So, $L(t)$ can be expressed as:

$$L(t) = M(t) \cup \{l_{\text{answer}}\}, \text{ where } t \in C_{\text{train}}$$

Instances - Testing

- ▶ During testing, for each token t with context C , we supply an instance (t, l, C) to the model for each label choice l for that token, and the model returns a score between 0 and 1 for each such instance.
- ▶ The label choice l which obtains the highest score in this manner is the choice predicted by our model for token t in context C .

Label Choices - Testing

- ▶ During testing, we are faced with a different challenge: since we do not know the correct label a priori, we must ensure that the set of label choices L for each token t will contain the correct choice as often as possible.
- ▶ Therefore, in addition to the set of choices $M(t)$ returned by SAMA, we also include in $L(t)$ the labels l observed for that token earlier in the training data C_{train} and in the global context C_{global} .

Label Choices - Testing

- ▶ This can be expressed as:

$$L(t) = M(t) \cup \{l \mid l \text{ was observed for } t \text{ in } C_{\text{train}}\} \cup \{l \mid l \text{ was observed for } t \text{ in } C_{\text{global}}\}, \text{ where } t \in C_{\text{test}}$$

Label Choices

- ▶ There is still no guarantee that the correct label will be contained in L for every token
- ▶ We find that this indeed the case for about 2% of the tokens in our test set C_{test} .
- ▶ For these tokens, the constraints of the available data imply that our model will not be able to find the correct answer regardless of the feature set or machine learning algorithm used.

Features

- ▶ We wish to determine features of text that are useful in predicting lemmas and morphological analyses.
- ▶ Consider the following string of tokens:

$\dots t_{-2} t_{-1} t t_{+1} t_{+2} \dots$

Let the initial and final letters of t_{-1} be denoted by li_{-1} and lf_{-1} respectively. Similarly, the initial and final letters of t_{+1} are li_{+1} and lf_{+1} respectively. Let l be one of the label choices for token t .

Features

- ▶ We use not only features of individual tokens, such as the frequency in C_{global} of $/$ for the token t , but also features of more than one tokens, for example, the frequency in C_{train} of $/$ for the token t in the instances when t is followed by the token t_{+1} .
- ▶ Features based on global context are more useful in predicting morphological analyses, while those based on local training data are more useful in predicting lemmas; consequently, we use both types.

Features

- ▶ Let F_{BASELINE} denote the set of all such basic features. We will use this set to create a baseline model. These features are described in detail in the following table.

Features - F_{BASELINE}

Feature	Description
$Freq_{global/train}$	Number of tokens in C_{train} or C_{global} whose label is l
$PrevWord_{global/train}$	Number of tokens in C_{train} or C_{global} for which label is l and previous token is t_{-1}
$NextWord_{global/train}$	Number of tokens in C_{train} or C_{global} for which label is l and following token is t_{+1}
$PreviLetter_{global/train}$	Number of tokens in C_{train} or C_{global} for which label is l and first letter of previous token is li_{-1}
$NextiLetter_{global/train}$	Number of tokens in C_{train} or C_{global} for which label is l and first letter of next token is li_{+1}
$PrevfLetter_{global/train}$	Number of tokens in C_{train} or C_{global} for which label is l and final letter of previous token is lf_{-1}
$NextfLetter_{global/train}$	Number of tokens in C_{train} or C_{global} for which label is l and final letter of next token is lf_{+1}

Features - F_{BASELINE}

- ▶ The set F_{BASELINE} can thus be expressed formally as:

$$F_{\text{BASELINE}} = \{ \text{Freq}_{\text{global}} \cup \text{PrevWord}_{\text{global}} \cup \text{NextWord}_{\text{global}} \cup \\ \text{PreviLetter}_{\text{global}} \cup \text{NextiLetter}_{\text{global}} \cup \text{PrevfLetter}_{\text{global}} \cup \\ \text{NextfLetter}_{\text{global}} \cup \text{Freq}_{\text{train}} \cup \text{PrevWord}_{\text{train}} \cup \\ \text{NextWord}_{\text{train}} \cup \text{PreviLetter}_{\text{train}} \cup \text{NextiLetter}_{\text{train}} \cup \\ \text{PrevfLetter}_{\text{train}} \cup \text{NextfLetter}_{\text{train}} \}$$

Morphosyntactic Features (MSFs)

- ▶ For each label l , we break down its morphological information into various morphosyntactic features (MSFs) such as basic part-of-speech, suffix, gender, number, mood, etc.
- ▶ The lemma of l is also a feature of l , and although it isn't a morphological concept, we shall refer to it as an MSF for the sake of convenience.
- ▶ A complete list of these MSFs and their possible values is detailed in the following table.

MSFs – Table of MSFs

MSF	Description	Possible Values
LEMMA	Lemma	Set of all Arabic lemmas
PRE1	1 st prefix	Conjunction, preposition, particle, NA
PRE2	2 nd prefix	Conjunction, preposition, particle, NA
DET	Determiner	Yes, No
POS	Basic part-of-speech	Noun, Adjective, Verb, etc.
DPOS	Addn. info. about POS	Proper (for Noun), Intransitive (for Verb), NA, etc.
SUF	Suffix	Yes, No
PERPOS	Person info. of POS	1, 2, 3, NA
NUMPOS	Number info. of POS	Singular, dual, plural, NA
GENPOS	Gender info. of POS	Male, Female, NA
PERSUF	Person info. of SUF	1, 2, 3, NA
NUMSUF	Number info. of SUF	Singular, dual, plural, NA
GENSUF	Gender info. of SUF	Male, Female, NA
MOOD	Mood	Indicative, subjunctive, jussive, NA
PRON	Pronoun clitic	Combined info of number, gender, person of pronoun, NA

MSFs – Examples of label values

- ▶ In the following table, we illustrate the values of these MSFs for some of labels mentioned earlier in our table of SAMA outputs

MSFs – Examples of label values

Label	Halam-u_1 + IV3MS + IV + IVSUFF_MOOD:I	tAliy_1 + PART + PREP + DET + ADJ	SafoHap_1 + DET + NOUN + NSUFF_FEM_SG
LEMMA	Halam-u_1	tAliy_1	SafoHap_1
PRE1	-	PREP	-
PRE2	-	PART	-
DET	-	Yes	Yes
POS	Verb	Adjective	Noun
DPOS	Intransitive	-	-
SUF	Yes	-	-
PERPOS	3 rd	-	-

MSFs – Examples of label values (continued)

Label	Halam-u_1 + IV3MS + IV + IVSUFF_MOOD:I	tAliy_1 + PART + PREP + DET + ADJ	SafoHap_1 + DET + NOUN + NSUFF_FEM_SG
NUMPOS	Singular	-	-
GENPOS	Masculine	-	-
PERSUF	-	-	-
NUMSUF	-	-	Singular
GENSUF	-	-	Feminine
MOOD	Indicative	-	-
PRON	-	-	-

MSFs - F_m

- ▶ For each MSF m we define a set of features F_m similar to F_{BASELINE} . However, these features are evaluated by the MSF m of a label l , not the entire label.
- ▶ Consider the MSF basic part-of-speech (POS). For this MSF, let us denote the feature analogous to $\text{Freq}_{\text{global}}$ as $\text{Freq_POS}_{\text{global}}$.
- ▶ Suppose the POS value of a label l is p ; $\text{Freq_POS}_{\text{global}}$ refers to the number of tokens in C_{global} whose labels have POS value p .

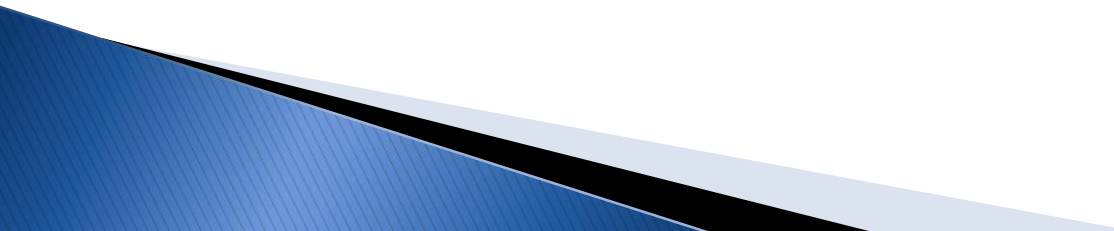
MSFs - F_m

- ▶ F_m can thus be expressed formally as:
- ▶ $F_m = \{Freq_m_{global} \cup PrevWord_m_{global} \cup NextWord_m_{global} \cup PreviLetter_m_{global} \cup NextiLetter_m_{global} \cup PrevfLetter_m_{global} \cup NextfLetter_m_{global} \cup Freq_m_{train} \cup PrevWord_m_{train} \cup NextWord_m_{train} \cup PreviLetter_m_{train} \cup NextiLetter_m_{train} \cup PrevfLetter_m_{train} \cup NextfLetter_m_{train}\}$

MSFs - $F_{\text{AGGREGATE}}$

- ▶ Finally, we define an feature set $F_{\text{AGGREGATE}}$ which includes not only all the features of F_{BASELINE} , but also all features of the feature sets F_m for each MSF m .
- ▶ $F_{\text{AGGREGATE}} = F_{\text{BASELINE}} \cup \{F_m \mid m \in \{\text{LEMMA, PRE1, PRE2, DET, POS, DPOS, SUF, PERPOS, NUMPOS, GENPOS, PERSUF, NUMSUF, GENSUF, MOOD, PRON}\}\}$
- ▶ $F_{\text{AGGREGATE}}$ is our final feature set. As we shall see in the following sections, we use $F_{\text{AGGREGATE}}$ to train an SVM model, and that is our final output model.

Results

- ▶ We use two metrics of accuracy: **Accuracy1** measures the percentage of tokens for which the model assigns the highest probability to the correct label or MSF value
 - ▶ **Accuracy2** measures the percentage of tokens for which the correct label or MSF value is one of the two highest ranked choices returned by the model.
- 

Results

- ▶ We perform 10-fold cross validation, forming C_{train} and C_{test} in a 7:3 proportion from Clocal, as described earlier.
- ▶ We train a SVM model SVM_m for each MSF m on C_{train} , using the feature set F_m . We test each such model on C_{test} . The results are in the following table.

Results – Individual MSFs

Model	Accuracy1	Model Accuracy2
SVM _{LEMMA}	.889	.951
SVM _{PRE1}	.981	.986
SVM _{PRE2}	.998	1
SVM _{DET}	.993	.999
SVM _{POS}	.766	.960
SVM _{DPOS}	.897	.981
SVM _{SUF}	.924	.975
SVM _{PERPOS}	.970	.999
SVM _{NUMPOS}	.968	.998
SVM _{GENPOS}	.982	.999
SVM _{PERSUF}	.968	.999
SVM _{NUMSUF}	.918	.995
SVM _{GENSUF}	.884	.996
SVM _{MOOD}	.984	.986
SVM _{PRON}	.982	.994

Results - SVM_{AGGREGATE}

- ▶ We also train a SVM model SVM_{AGGREGATE} which utilizes the feature set $F_{AGGREGATE}$, on the training dataset C_{train} . This is our output model, and its results are summarized in the following table.

Model	Accuracy1	Accuracy2
SVM _{AGGREGATE}	.906	.962

Results - SVM_{AGGREGATE}

- ▶ The accuracy that we are thus able to achieve using the SVM_{AGGREGATE} model is higher than prior approaches have been able to achieve so far for combined morphological and word sense disambiguation.
- ▶ As we saw earlier, for about 2% of the tokens, the correct label is simply not contained in the set of choices considered.
- ▶ Excluding these tokens, the **Accuracy2** score of SVM_{AGGREGATE} rises to more than 98%.

Results - Speed

- ▶ The SVM_{AGGREGATE} model is able to process Arabic text at a speed of approximately thirty tokens per second on a reasonably fast workstation.

Alternative Approaches - Baseline

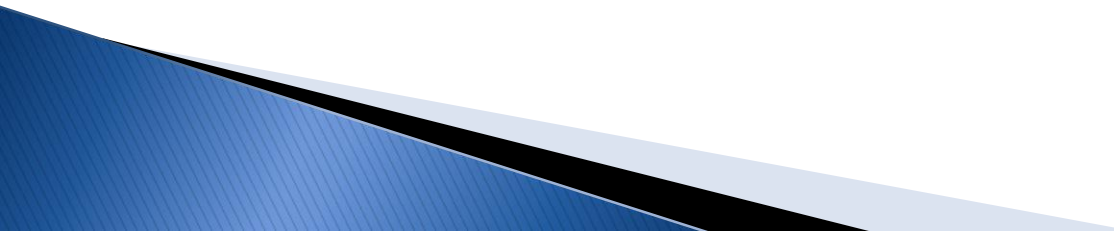
- ▶ To illustrate the benefit obtained by breaking down each label / into MSFs, we train on C_{train} a SVM model $\text{SVM}_{\text{BASELINE}}$ based on the feature set F_{BASELINE} , which contains features of entire labels only, not individual MSFs. Its results are tabulated in Table 8. As we can see, $\text{SVM}_{\text{AGGREGATE}}$ performs significantly better in terms of both **Accuracy1** and **Accuracy2** scores.

Model	Accuracy1	Accuracy2
$\text{SVM}_{\text{BASELINE}}$.864	.909
$\text{SVM}_{\text{AGGREGATE}}$.906	.962

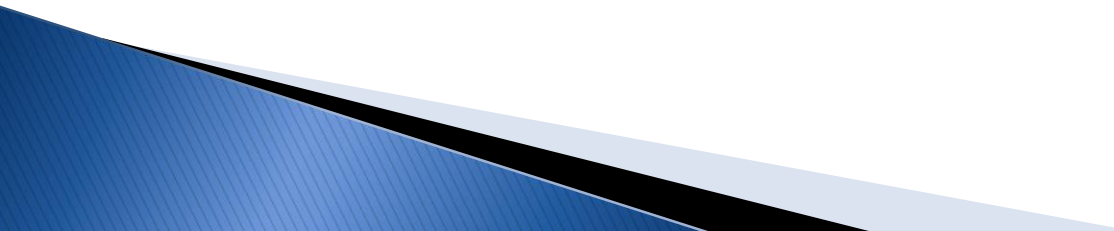
Alternative Approaches

- ▶ We consider whether it might not be better to build models for each MSF based on conditional random fields, instead of SVMs.
- ▶ To test the feasibility of this idea, we build simple two pass approximations to CRFs for each MSF, in which the correct label choices for the previous and following token of a given token are used as features in the second pass.
- ▶ We find very little statistical improvement
- ▶ Much more time required to train and test a CRF model

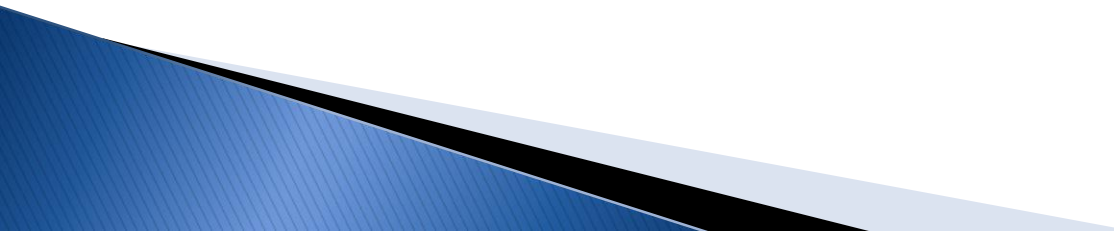
Implementation

- ▶ The SVM_{AGGREGATE} model, like the SAMA morphological analyzer, is written in Perl.
 - ▶ We use the LibSVM library to implement support vector machines.
 - ▶ We refer to this implementation of our model as the Standard Arabic Morphological Tagger (SAMT).
- 

Applications

- ▶ Arabic corpus annotation has so far relied on using a morphological analyzer such as SAMA to generate various morphology and lemma choices, and supplying these to manual annotators who then pick the correct choice.
 - ▶ However, a major constraint is that comprehensive morphological analyzers such as SAMA can generate dozens of choices for each word, each of which must be examined by the annotator.
 - ▶ Moreover, morphological analyzers do not provide any information about the likelihood of a particular choice being correct.
- 

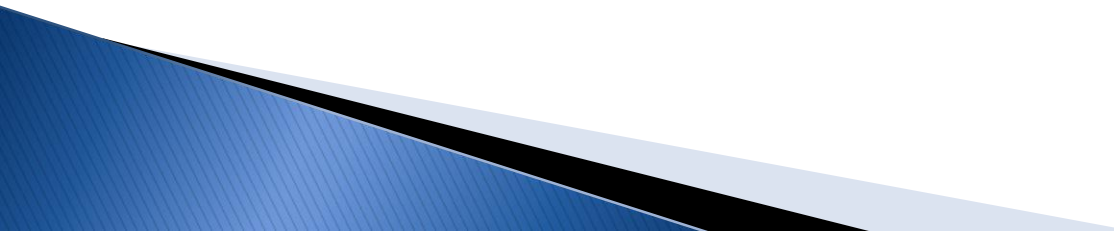
Applications – Corpus Annotation

- ▶ Using our model to rank these choices in order of their probabilities ensures that for the vast majority of tokens, the manual annotators need only consider a couple of choices.
 - ▶ We believe the use of our model will allow us to accelerate the rate of annotation by a factor of two.
- 

Applications – Corpus Annotation

- ▶ As the accuracy of the model increases with further research, we can envision the possibility of completely automating the annotation process, and thus making it even faster and easier to obtain new annotated corpora.

Applications - Language Helpers

- ▶ Our model is also able to aid intermediate Arabic language learners.
 - ▶ We have created an interface that allows users to submit news articles online, and e-mails labeled versions of those articles back to users.
 - ▶ In addition, we have created a web interface to our model that allows users to input sentences or paragraphs of text and obtain annotations for that text within a few seconds.
- 

Applications – Screenshot of SAMT

Paste in either UTF-8 or Buckwalter encoded Arabic strings

صديق يقول أنه كان يعلم ستة أشهر بالعراقي يعني
 الانضباطية يركضون وراة ست شهر وهو خارج العراق
 لدا لا يمكن أن ننسى جرائم النظام التي ارتكبها قبل
 التاسع من نيسان

Indicate which encoding you are using: UTF-8 Buckwalter

Submit Query

The text you submitted in UTF-8 encoding is tokenized and analyzed as follows:

Tkn#	Word	Top-Choice	Top-Score	2nd-Choice	2nd-Score
1	صديق	Sadiyq_1:Sadiyq/NOUN <i>friend</i>	0.991	Sadiyq_1:Sadiyq/ADJ <i>friend</i>	0.285
2	يقول	qAl-u_1.ya/TV3MS+quwl/TV+u/TVSUFF_MOOD:I <i>be said,said,say</i>	0.351	qAl-u_1.ya/TV3MS+quwl/TV+a/TVSUFF_MOOD:S <i>be said,said,say</i>	0.113
3	أنه	>an~a_1:>an~a/SUB_CONJ+huw/PRON_3MS <i>that</i>	0.890	>an~ap_1:>an~/NOUN+ap/NSUFF_FEM_SG <i>moan;complaint</i>	0.044
4	كان	kAn-u_1.kAn/PV+a/PVSUFF_SUBJ:3MS <i>be;if,whether + was,were</i>	0.752	n_1.ka/PREP+ n/NOUN <i>time;moment</i>	-0.067
5	يحلم	Halam-u_1.ya/TV3MS+Holum/TV+u/TVSUFF_MOOD:I <i>dream</i>	0.386	Halam-u_1.ya/TV3MS+Holum /TV+a/TVSUFF_MOOD:S <i>dream</i>	0.137


Related Work

- ▶ Hajič shows that for highly inflectional languages, the use of a morphological analyzer improves accuracy of disambiguation.
- ▶ Diab et al. perform tokenization, POS tagging and base phrase chunking using a SVM based learner; however, they do not use a morphological analyzer.
- ▶ Ahmed and Nürnberger perform word-sense disambiguation using a Naïve Bayesian classifier and rely on parallel corpora and matching schemes instead of a morphological analyzer

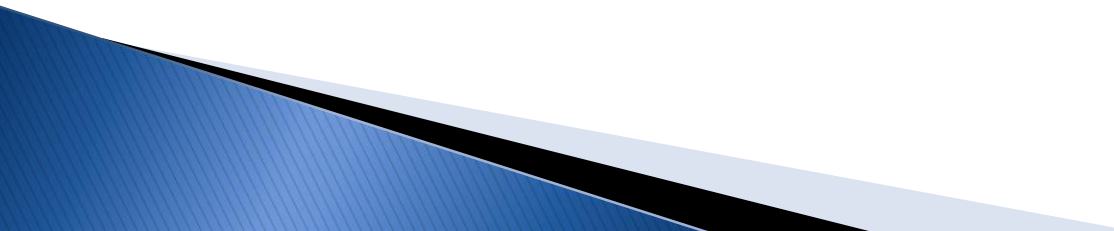
Related Work - MADA

- ▶ Habash, Rambow et al. provide an algorithm to perform morphological disambiguation using a morphological analyzer; however, this algorithm does not include lemmatization.
- ▶ Roth, Rambow, Habash et al. later show how this algorithm can be adapted to also perform lemmatization; they call their system MADA.
- ▶ However, our work differs from theirs in a number of respects

Related Work - MADA

- ▶ Firstly, they use features of only individual tokens for morphological analysis while we use features of individual tokens as well as of bigrams.
 - ▶ They also use only local context to obtain feature values, while we use both local and global context.
 - ▶ Also, they do not learn a single model from an aggregate feature set, rather they combine models for individual MSFs by using weighted agreement.
- 

Related Work - MADA

- ▶ Finally, they rely solely on a morphological analyzer for label choices, while we make use of global context in addition to a morphological analyzer.
 - ▶ This helps minimize the number of cases where the set of choices does not contain the correct label.
 - ▶ Consequently, we are able to achieve higher overall scores of accuracy than those reported by them.
- 

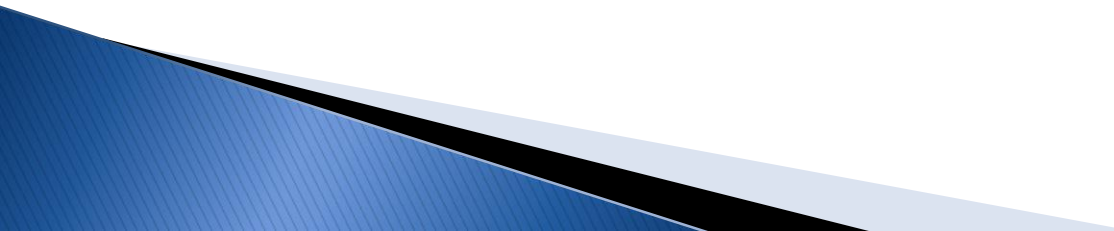
Related work - Comparison to MADA

- ▶ We obtained a copy of the MADA system in order to measure its performance with respect to SAMT.
- ▶ We used their full feature set, and performed 10-fold cross validation on the same testing data that we used to test SAMT.
- ▶ MADA gives an Accuracy₁ score of 83.1%, compared to 90.6% obtained by SAMT.

Conclusion

- ▶ We showed that a machine learning approach that utilizes both local and global context can perform simultaneous lemmatization and morphological analysis with more accuracy than existing algorithms.
- ▶ We also showed that a model based on an aggregate feature set of all MSFs works better than a feature set that does not break down labels into individual MSFs.
- ▶ Finally, we showed that using features of n -grams in addition to those of individual tokens is preferable to using features of only individual tokens.

Future Directions

- ▶ We believe the approach we have used to create the SAMT system can be successfully applied to not just Arabic but also other similar languages that have highly inflectional morphology.
 - ▶ Also, increasing the accuracy of SAMT from its current score of 96% to a score of 99% or above would effectively allow the automation of the process of annotating Arabic text that is currently performed manually.
- 

Acknowledgements

- ▶ We would like to thank David Graff, Seth Kulick and Basma Bouziri for their help and contribution.
- 

References

- ▶ Ryan Roth, Owen Rambow, Nizar Habash, Mona Diab, and Cynthia Rudin. 2008. Arabic Morphological Tagging, Diacritization, and Lemmatization Using Lexeme Models and Feature Ranking. In ACL'08, Columbus, Ohio, USA.
- ▶ Nizar Habash and Owen Rambow. 2005. Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In ACL'05, Ann Arbor, MI, USA.
- ▶ Farag Ahmed and Andreas Nürnberger. 2008. Arabic/English Word Translation Disambiguation using Parallel Corpora and Matching Schemes. In EAMT conference, Hamburg, Germany.

References

- ▶ Mohamed Maamouri, David Graff, Basma Bouziri, Sondos Krouna, Seth Kulick. 2009. LDC Standard Arabic Morphological Analyzer (SAMA) v. 3.0. LDC Catalog No.: LDC2009E44. Special GALE release to be followed by a full LDC publication.
- ▶ Tim Buckwalter. 2004. Buckwalter Arabic Morphological Analyzer version 2.0. LDC Catalog No.: LDC2004L02
- ▶ Mohamed Maamouri, Ann Bies, and Tim Buckwalter. 2004. The Penn Arabic Treebank: Building a large scale annotated Arabic corpus. In NEMLAR Conference on Arabic Language Resources and Tools, Cairo, Egypt.

References

- ▶ Chih-Chung Chang and Chih-Jen Lin. 2001. LIBSVM: a library for support vector machines, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
- ▶ Mona Diab, Kadri Hacioglu, and Daniel Jurafsky. 2004. Automatic tagging of Arabic text: From raw text to base phrase chunks. In 5th Meeting of the North American Chapter of the Association for Computational Linguistics/Human Language Technologies Conference (HLT-NAACL04), Boston, MA.
- ▶ Monica Rogati, J. Scott McCarley, and Yiming Yang. 2003. Unsupervised learning of Arabic stemming using a parallel corpus. In 41st Meeting of the Association for Computational Linguistics (ACL'03), pages 391–398, Sapporo, Japan.

References

- ▶ Jan Hajič, Otakar Smrč, Tim Buckwalter, and Hubert Jin. 2005. Feature-based tagger of approximations of functional Arabic morphology. In Proceedings of the Workshop on Treebanks and Linguistic Theories (TLT), Barcelona, Spain.
- ▶ Jan Hajič. 2000. Morphological tagging: Data vs. dictionaries. In 1st Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL'00), Seattle, WA.
- ▶ Imad A. Al-Sughaiyer and Ibrahim A. Al-Kharashi. 2004. Arabic morphological analysis techniques: A comprehensive survey. *Journal of the American Society for Information Science and Technology*, 55(3):189–213.

Thank You

