CATiB: The Columbia Arabic Treebank

Nizar Habash and Ryan M. Roth
Center for Computational Learning Systems
Columbia University, New York, USA
{habash,ryanr}@ccls.columbia.edu

Abstract

The Columbia Arabic Treebank (CATiB) is a database of syntactic analyses of Arabic sentences. CATiB contrasts with previous approaches to Arabic treebanking in its emphasis on speed with some constraints on linguistic richness. Two basic ideas inspire the CATiB approach: no annotation of redundant information and using representations and terminology inspired by traditional Arabic syntax. We describe CATiB's representation and annotation procedure, and report on inter-annotator agreement and speed.

1 Introduction and Motivation

Treebanks are collections of manually-annotated syntactic analyses of sentences. They are primarily intended for building models for statistical parsing; however, they are often enriched for general natural language processing purposes. For Arabic, two important treebanking efforts exist: the Penn Arabic Treebank (PATB) (Maamouri et al., 2004) and the Prague Arabic Dependency Treebank (PADT) (Smrž and Hajič, 2006). In addition to syntactic annotations, both resources are annotated with rich morphological and semantic information such as full part-of-speech (POS) tags, lemmas, semantic roles, and diacritizations. This allows these treebanks to be used for training a variety of applications other than parsing, such as tokenization, diacritization, POS tagging, morphological disambiguation, base phrase chunking, and semantic role labeling.

In this paper, we describe a new Arabic treebanking effort: the Columbia Arabic Treebank (CATiB).[1] CATiB is motivated by the following three observations. First, as far as parsing Arabic research, much of the non-syntactic rich annotations are not used. For example, PATB has over 400 tags, but they are typically reduced to around 36 tags in training and testing parsers (Kulick et al., 2006). The reduction addresses the fact that sub-tags indicating case and other similar features are essentially determined syntactically and are hard to automatically tag accurately. Second, under time restrictions, the creation of a treebank faces a tradeoff between linguistic richness and treebank size. The richer the annotations, the slower the annotation process, the smaller the resulting treebank. Obviously, bigger treebanks are desirable for building better parsers. Third, both PATB and PADT use complex syntactic representations that come from modern linguistic traditions that differ from Arabic's long history of syntactic studies. The use of these representations puts higher requirements on the kind of annotators to hire and the length of their initial training.

CATiB contrasts with PATB and PADT in putting an emphasis on annotation speed for the specific task of parser training. Two basic ideas inspire the CATiB approach. First, CATiB avoids annotation of redundant linguistic information or information not targeted in current parsing research. For example, nominal case markers in Arabic have been shown to be automatically determinable from syntax and word morphology and needn't be manually annotated (Habash et al., 2007a). Also, phrasal co-indexation, empty pronouns, and full lemma disambiguation are not currently used in parsing research so we do not include them in CATiB. Second, CATiB uses a simple intuitive dependency representation and terminology inspired by Arabic's long tradition of syntactic studies. For example, CATiB relation labels include *tamyiz* (specification) and *idafa* (possessive construction) in addition to universal predicate-argument structure labels such as *subject*, *object* and *modifier*. These representation choices make it easier to train annotators without being restricted to hire people who have degrees in linguistics.

This paper briefly describes CATiB's representation and annotation procedure, and reports on produced data, achieved inter-annotator agreement and annotation speeds.

[1]This work was supported by Defense Advanced Research Projects Agency Contract No. HR0011-08-C-0110.

Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, pages 221–224,
Suntec, Singapore, 4 August 2009.

2 CATiB: Columbia Arabic Treebank

CATiB uses the same basic tokenization scheme used by PATB and PADT. However, the CATiB POS tag set is much smaller than the PATB's. Whereas PATB uses over 400 tags specifying every aspect of Arabic word morphology such as definiteness, gender, number, person, mood, voice and case, CATiB uses 6 POS tags: **NOM** (non-proper nominals including nouns, pronouns, adjectives and adverbs), **PROP** (proper nouns), **VRB** (active-voice verbs), **VRB-PASS** (passive-voice verbs), **PRT** (particles such as prepositions or conjunctions) and **PNX** (punctuation).[2]

CATiB's dependency links are labeled with one of eight relation labels: **SBJ** (subject of verb or topic of simple nominal sentence), **OBJ** (object of verb, preposition, or deverbal noun), **TPC** (topic in complex nominal sentences containing an explicit pronominal referent), **PRD** (predicate marking the complement of the extended copular constructions for *kAn*[3] كان واخواتها and *An* ان واخواتها), **IDF** (relation between the possessor [dependent] to the possessed [head] in the idafa/possesive nominal construction), **TMZ** (relation of the specifier [dependent] to the specified [head] in the tamyiz/specification nominal constructions), **MOD** (general modifier of verbs or nouns), and **—** (marking *flatness* inside constructions such as first-last proper name sequences). This relation label set is much smaller than the twenty or so dashtags used in PATB to mark syntactic and semantic functions. No empty categories and no phrase co-indexation are made explicit. No semantic relations (such as time and place) are annotated.

Figure 1 presents an example of a tree in CATiB annotation. In this example, the verb زاروا *zArwA* 'visited' heads a subject, an object and a prepositional phrase. The subject includes a complex number construction formed using idafa and tamyiz and headed by the number خمسون *xmswn* 'fifty', which is the only carrier of the subject's syntactic nominative case here. The preposition في *fy* heads the prepositional phrase, whose object is a proper noun, تموز *tmwz* 'July' with an adjectival modifier, الماضي *AlmADy* 'last'. See Habash et al. (2009) for a full description of CATiB's guidelines and a detailed comparison with PATB and PADT.

[2]We are able to reproduce a parsing-tailored tag set [size 36] (Kulick et al., 2006) automatically at 98.5% accuracy using features from the annotated trees. Details of this result will be presented in a future publication.

[3]Arabic transliterations are in the Habash-Soudi-Buckwalter transliteration scheme (Habash et al., 2007b).

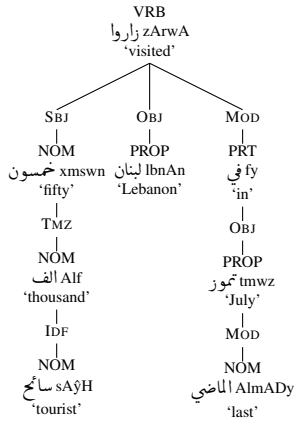

Figure 1: CATiB annotation for the sentence خمسون الف سائح زاروا لبنان في تموز الماضي
xmswn Alf sAŷH zArwA lbnAn fy tmwz AlmADy
'50 thousand tourists visited Lebanon last July.'

3 Annotation Procedure

Although CATiB is independent of previous annotation projects, it builds on existing resources and lessons learned. For instance, CATiB's pipeline uses PATB-trained tools for tokenization, POS-tagging and parsing. We also use the TrEd annotation interface developed in coordination with the PADT. Similarly, our annotation manual is guided by the wonderfully detailed manual of the PATB for coverage (Maamouri et al., 2008).

Annotators Our five annotators and their supervisor are all educated native Arabic speakers. Annotators are hired on a part-time basis and are not required to be on-site. The annotation files are exchanged electronically. This arrangement allows more annotators to participate, and reduces logistical problems. However, having no full-time annotators limits the overall weekly annotation rate. Annotator training took about two months (150 hrs/annotator on average). This training time is much shorter than the PATB's six-month training period.[4]

Below, we describe our pipeline in some detail including the different resources we use.

Data Preparation The data to annotate is split into *batches* of 3-5 documents each, with each document containing around 15-20 sentences (400-600 tokens). Each annotator works on one batch at a time. This procedure and the size of the batches was determined to be optimal for both the software and the annotators' productivity. To track the annotation quality, several key documents are selected for inter-annotator agreement (IAA) checks. The IAA documents are chosen to

[4]Personal communication with Mohamed Maamouri.

cover a range of sources and to be of average document size. These documents (collectively about 10% of the token volume) are seeded throughout the batches. Every annotator eventually annotates each one of the IAA documents, but is never told which documents are for IAA.

Automatic Tokenization and POS Tagging We use the MADA&TOKAN toolkit (Habash and Rambow, 2005) for initial tokenization and POS tagging. The tokenization F-score is 99.1% and the POS tagging accuracy (on the CATiB POS tag set; with gold tokenization) is above 97.7%.

Manual Tokenization Correction Tokenization decisions are manually checked and corrected by the annotation supervisor. New POS tags are assigned manually only for corrected tokens. Full POS tag correction is done as part of the manual annotation step (see below). The speed of this step is well over 6K tokens/hour.

Automatic Parsing Initial dependency parsing in CATiB is conducted using MaltParser (Nivre et al., 2007). An initial parsing model was built using an automatic constituency-to-dependency conversion of a section of PATB *part 3* (PATB3-Train, 339K tokens). The quality of the automatic conversion step is measured against a hand-annotated version of an automatically converted held-out section of PATB3 (PATB3-Dev, 31K tokens). The results are 87.2%, 93.16% and 83.2% for attachment (**ATT**), label (**LAB**) and labeled attachment (**LABATT**) accuracies, respectively. These numbers are 95%, 98% and 94% (respectively) of the IAA scores on that set.[5] At the production midpoint another parsing model was trained by adding all the CATiB annotations generated up to that point (513K tokens total). An evaluation of the parser against the CATiB version of PATB3-Dev shows the **ATT**, **LAB** and **LABATT** accuracies are 81.7%, 91.1% and 77.4% respectively.[6]

Manual Annotation CATiB uses the TrEd tool as a visual interface for annotation.[7] The parsed trees are converted to TrEd format and delivered to the annotators. The annotators are asked to only correct the POS, syntactic structure and relation labels. Once annotated (i.e. corrected), the documents are returned to be packaged for release.

[5]Conversion will be discussed in a future publication.

[6]Since CATiB POS tag set is rather small, we extend it automatically deterministically to a larger tag set for parsing purposes. Details will be presented in a future publication.

[7]http://ufal.mff.cuni.cz/∼pajas/tred

IAA Set	Sents	POS	ATT	LAB	LABATT
PATB3-Dev	All	98.6	91.5	95.3	88.8
	≤ 40	98.7	91.7	94.7	88.6
PROD	All	97.6	89.2	93.0	85.0
	≤ 40	97.7	91.5	94.1	87.7

Table 1: Average pairwise IAA accuracies for 5 annotators. The **Sents** column indicates which sentences were evaluated, based on token length. The sizes of the sets are 2.4K (PATB3-Dev) and 3.8K (PROD) tokens.

4 Results

Data Sets CATiB annotated data is taken from the following LDC-provided resources:[8] LDC2007E46, LDC2007E87, GALE-DEV07, MT05 test set, MT06 test set, and PATB (part 3). These datasets are 2004-2007 newswire feeds collected from different news agencies and news papers, such as Agence France Presse, Xinhua, Al-Hayat, Al-Asharq Al-Awsat, Al-Quds Al-Arabi, An-Nahar, Al-Ahram and As-Sabah. The CATiB-annotated PATB3 portion is extracted from An-Nahar news articles from 2002. Headlines, datelines and bylines are not annotated and some sentences are excluded for excessive (>300 tokens) length and formatting problems. Over 273K tokens (228K words, 7,121 trees) of data were annotated, not counting IAA duplications. In addition, the PATB part 1, part 2 and part 3 data is automatically converted into CATiB representation. This converted data contributes an additional 735K tokens (613K words, 24,198 trees). Collectively, the CATiB version 1.0 release contains over 1M tokens (841K words, 31,319 trees), including annotated and converted data.

Annotator Speeds Our POS and syntax annotation rate is 540 tokens/hour (with some reaching rates as high as 715 tokens/hour). However, due to the current part-time arrangement, annotators worked an average of only 6 hours/week, which meant that data was annotated at an average rate of 15K tokens/week. These speeds are much higher than reported speeds for complete (POS+syntax) annotation in PATB (around 250-300 tokens/hour) and PADT (around 75 tokens/hour).[9]

Basic Inter-Annotator Agreement We present IAA scores for **ATT**, **LAB** and **LABATT** on IAA

[8]http://www.ldc.upenn.edu/

[9]Extrapolated from personal communications, Mohamed Maamouri and Otakar Smrž. In the PATB, the syntactic annotation step alone has similar speed to CATiB's full POS and syntax annotation. The POS annotation step is what slows down the whole process in PATB.

IAA File	Toks/hr	POS	ATT	LAB	LABATT
HI	398	97.0	94.7	96.1	91.2
HI-S	956	97.0	97.8	97.9	95.7
LO	476	98.3	88.8	91.7	82.3
LO-S	944	97.7	91.0	93.8	85.8

Table 2: Highest and lowest average pairwise IAA accuracies for 5 annotators achieved on a single document – before and after serial annotation. The "-S" suffix indicates the result after the second annotation.

subsets from two data sets in Table 1: PATB3-Dev is based on an automatically converted PATB set and PROD refers to all the new CATiB data. We compare the IAA scores for all sentences and for sentences of token length ≤ 40 tokens. The IAA scores in PROD are lower than PATB3-Dev, this is understandable given that the error rate of the conversion from a manual annotation (starting point of PATB3-Dev) is lower than parsing (starting point for PROD). Length seems to make a big difference in performance for PROD, but less so for PATB3-Dev, which makes sense given their origins. Annotation training did not include very long sentences. Excluding long sentences during production was not possible because the data has a high proportion of very long sentences: for PROD set, 41% of sentences had >40 tokens and they constituted over 61% of all tokens.

The best reported IAA number for PATB is 94.3% F-measure after extensive efforts (Maamouri et al., 2008). This number does not include dashtags, empty categories or indices. Our numbers cannot be directly compared to their number because of the different metrics used for different representations.

Serial Inter-Annotator Agreement We test the value of *serial annotation*, a procedure in which the output of annotation is passed again as input to another annotator in an attempt to improve it. The IAA documents with the highest (HI, 333 tokens) and lowest (LO, 350 tokens) agreement scores in PROD are selected. The results, shown in Table 2, indicate that serial annotation is very helpful reducing **LABATT** error by 20-50%. The reduction in LO is not as large as that in HI, unfortunately. The second round of annotation is almost twice as fast as the first round. The overall reduction in speed (end-to-end) is around 30%.

Disagreement Analysis We conduct an error analysis of the basic-annotation disagreements in HI and LO. The two sets differ in sentence length, source and genre: HI has 28 tokens/sentence and contains AFP general news, while LO has 58 tokens/sentence and contains Xinhua financial news. The most common POS disagreement in both sets is NOM/PROP confusion, a common issue in Arabic POS tagging in general. The most common attachment disagreements in LO are as follows: prepositional phrase (PP) and nominal modifiers (8% of the words had at least one dissenting annotation), complex constructions (dates, proper nouns, numbers and currencies) (6%), subordination/coordination (4%), among others. The respective proportions for HI are 5%, 5% and 1%. Label disagreements are mostly in nominal modification (MOD/TMZ/IDF/—) (LO 10%, HI 5% of the words had at least one dissenting annotation).

The error differences between HI and LO seem to primarily correlate with length difference and less with genre and source differences.

5 Conclusion and Future Work

We presented CATiB, a treebank for Arabic parsing built with faster annotation speed in mind. In the future, we plan to extend our annotation guidelines focusing on longer sentences and specific complex constructions, introduce serial annotation as a standard part of the annotation pipeline, and enrich the treebank with automatically generated morphological information.

References

N. Habash, R. Faraj and R. Roth. 2009. Syntactic Annotation in the Columbia Arabic Treebank. In *Conference on Arabic Language Resources and Tools*, Cairo, Egypt.

N. Habash and O. Rambow. 2005. Arabic Tokenization, Part-of-Speech Tagging and Morphological Disambiguation in One Fell Swoop. In *ACL'05*, Ann Arbor, Michigan.

N. Habash, R. Gabbard, O. Rambow, S. Kulick, and M. Marcus. 2007a. Determining case in Arabic: Learning complex linguistic behavior requires complex linguistic features. In *EMNLP'07*, Prague, Czech Republic.

N. Habash, A. Soudi, and T. Buckwalter. 2007b. On Arabic Transliteration. In A. van den Bosch and A. Soudi, editors, *Arabic Computational Morphology*. Springer.

S. Kulick, R. Gabbard, and M. Marcus. 2006. Parsing the Arabic Treebank: Analysis and Improvements. In *Treebanks and Linguistic Theories Conference*, Prague, Czech Republic.

M. Maamouri, A. Bies, and T. Buckwalter. 2004. The Penn Arabic Treebank: Building a large-scale annotated Arabic corpus. In *Conference on Arabic Language Resources and Tools*, Cairo, Egypt.

M. Maamouri, A. Bies and S. Kulick. 2008. Enhancing the Arabic treebank: a collaborative effort toward new annotation guidelines. In *LREC'08*, Marrakech, Morocco.

J. Nivre, J. Hall, J. Nilsson, A. Chanev, G. Eryigit, S. Kubler, S. Marinov, and E. Marsi. 2007. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135.

O. Smrž and J. Hajič. 2006. The Other Arabic Treebank: Prague Dependencies and Functions. In Ali Farghaly, editor, *Arabic Computational Linguistics*. CSLI Publications.