MATAR: Morphology-based Tagger for Arabic

Fadi A. Zaraket Ameen Jaber
Department of Electrical and Computer Engineering
American University of Beirut
Email: {fz11,aaj15}@aub.edu.lb

***Abstract*—Computational linguistic and natural language processing automation tasks require text annotated with tags that represent the desired output of the task. The annotation tags serve for training, validation, and evaluation. Arabic morphological analysis, and tags associated with it such as part of speech and gloss tags, is key to Arabic computational linguistics and natural language processing. Several manual and automated tagging tools exist for text. Very few exist that are based on Arabic morphological analysis.**

In this paper, we present an open source tagging tool with visual interface that enables the construction of annotated Arabic text corpora with automatic morphology-based tags. The tool allows the specification of tags with Boolean formulae where the atomic predicates are *match* and *contain* relations between the morphological solution of part of the text and the value of a morphological feature. The tool allows the user to directly enter manual tags, to edit existing tags through a tag sensitive coloring interface, to compare tag sets, and compute accuracy results.

Keywords: Tagging; Arabic; Natural language processing; Morphological analysis; Computational linguistics.

I. Introduction

Computational linguistic (CL) and natural language processing (NLP) tasks consider text documents and map the text, or parts of it, to an output domain. *Machine translation* (MT), for example, maps text to sentences in a second language. *Information extraction* (IE) maps the text to entity categories. Supervised learning techniques require training text that is annotated with correct results to learn a computational model. Supervised and unsupervised techniques require reference and testing text that is annotated with correct results to evaluate the accuracy of the technique [12], [9], [18].

An *annotation*, referred to as *tag* in the sequel, relates a chunk of text to a *label*, or a *tag type*, that denotes a semantic value of interest to the NLP task.

Morphological analysis (MA) is key to Arabic CL and NLP even for simple tasks such as tokenization and stemming due to the morphological richness of the Arabic language and due to other problems such as missing short vowels, also known as diacritics, which are usually omitted in text and inferred by human readers [6]. Given an Arabic word delimited by white space and punctuation, MA returns the internal structure of the word composed of several morphemes including *affixes* (*prefixes* and *suffixes*), and *stems* [1]. A morphological analyzer returns a set of morphological solution vectors with features such as prefix, stem, suffix, part of speech (POS), *gloss*, and category tags. For example, Table I shows a sample

TABLE I. Solution vector for يَأْكُله *yaʾakulh* .

	Prefix	**Stem**	**Suffix**
Form	يَ *ya*	أْكُل *ʾakul*	ه *h*
POS tag	IV3MS+	VERB_IMPERFECT	IVSUFF_DO:3MS
Gloss tag	he/it	eat/consume	him/it

morphological solution for the word يَأْكُله *yaʾakulh* [1]. The POS tags `IV3MS` and `IVSUFF_DO:3MS` indicate a third person singular masculine subject pronoun attached to a verb, and a third person singular masculine object pronoun attached to an action verb, respectively. The verb has `VERB_IMPERFECT` as a POS tag.

In this paper, we present an open source morphology-based automatic tagger for Arabic (MATAR) that allows the specification of tag types with Boolean formulae over morphological features . Given a chunk of text t, Sarf, an in-house morphological analyzer [19], returns a set of morphological solutions $M(t) = \{m_1, m_2, \ldots, m_N\}$ where each solution is a vector of morphological features $m_i = \langle p, s, x, P, G, C\rangle, 1 \leq i \leq N$ where p, s and x are the prefix, stem, and suffix and P, G and C are the POS, gloss, and category tags, respectively. The atomic terms in a tag type formula consist of *identity* or *containment* predicates that relate the morphology solution vector to a morphological feature. Informally, identity denotes an exact match of the morphological feature, and containment denotes the existence of the feature in the solution vector.

Given a sequence of text elements $T = \langle t_1, t_2, \ldots t_M\rangle$, and a set of tag types $\mathcal{T}$ with their formulae, MATAR uses Sarf to compute the tags $R \subseteq T \times \mathcal{T}$. MATAR displays the results to the user using a visual interface with tag type sensitive coloring. MATAR allows the user to manually edit the result and build the corrected corpora. MATAR computes accuracy measures such as inter annotation agreement, and precision and recall, by comparing sets of tags and tag types. We used these capabilities of MATAR to rapidly build reference corpora and report accuracy results for several tasks [11], [20], [21].

MATAR has the following advantages.

- MATAR provides a novel and intuitive visual interface to build Boolean formulae over morphological features and thereafter compute automatic tags.
- MATAR allows the user to visually build category tag sets based on morphology features. The categories can later be used to define atomic terms in the formulae.
- MATAR provides the user with the ability to rapidly create annotated Arabic text corpora with sophisticated

[1]In this document, we use the default ArabTeX transliteration style ZDMG.

morphology based tags.

II. Related Work

The work in [5] presents an online supervised collaborative effort towards morphological and syntactic annotation of the Quran. The work in [4] presents a framework for interlingual annotation of parallel text corpora with multi-level representations. An overview of annotation tools and their Arabic-English word alignment issues concludes with a set of rules and guidelines needed in an Arabic annotation alignment tool [7]. The work in [8] presents the integration of the Standard Arabic Morphological Analyzer (SAMA) into the annotation workflow of the Arabic Treebank. Such tasks motivated us to build MATAr, a morphology based open source annotation tool for the Arabic language.

MMAX2 is a manual multi-level linguistic annotation tool with an XML based data model [14]. It enables the user to create, browse, visualize, and query annotations and may be able to resolve co-reference tags. BRAT is a multi-lingual user friendly manual web-based annotator that allows the construction of entity and relation annotation corpora [17]. BRAT provides an API for automated annotators to provide annotations. WordFreak is similar to BRAT. It supports Arabic text and can be extended through a plug-in architecture to integrate with NLP and CL tasks. The plug-in API may enable the use of automatic annotators along with customized visualization and annotation specifications [13]. AGTK is a toolkit for the development of text and speech annotation tools [10]. It provides import APIs from other data and graphical user interface (GUI) components. The work in [16] presents the extension of TrEd, a customizable general purpose tree editor, with the Arabic MorphoTrees annotation. The MorphoTrees present the morphological analyses in a hierarchical organization based on common features. MATAr differs from MMAX2, BRAT, WordFreak, AGTK, and TrEd in that it allows the user to specify sophisticated tag types using Boolean formulae of Arabic morphological features.

Fassieh is a commercial Arabic text annotation tool that enables the production of large Arabic text corpora [3]. The tool supports Arabic text factorizations including morphological analysis, POS tagging, full phonetic transcription, and lexical semantics analysis in an automatic mode. Fassieh is not directly accessible to the research community and requires commercial licensing. MATAr is open source and differs in that it allows the user to build tag types using Boolean formulae of several atomic terms.

Task specific annotation tools such as [2] uses enunciation semantic maps to automatically annotate directly reported Arabic and French speech. AraTation is another task specific tool for semantic annotation of Arabic news using web ontology based semantic maps [15]. We differ in that MATAr is general, and not task specific, and it uses morphology based features as atomic terms.

III. MATAr

MATAr takes a sequence of Arabic words $T = \langle t_1, t_2, \ldots, t_M \rangle$ as input text, and a set of tag types $\mathcal{T}$ with their formulae. MATAr is integrated with *Sarf*, an in-house open source Arabic morphological analyzer based on finite state transducers [19]. MATAr uses Sarf to compute a set of morphological solutions $M(t_i) = \{m_1, m_2, \ldots, m_N\}$ for each word $t_i, 1 \leq i \leq M$. Let $\mathcal{F} = \{\mathcal{P}, \mathcal{S}, \mathcal{X}, POS, GLOSS, CAT\}$ be the set of prefix, stem, suffix, POS, gloss, and abstract category tags in Sarf, respectively. Each morphological solution m is of the form $\langle p, s, x, P, G, C \rangle$ where $p \in \mathcal{P}, s \in \mathcal{S}, x \in \mathcal{X}, P \in POS, G \in GLOSS$, and $C \in CAT$.

Atomic terms: Let $\mathcal{O} = \{isA, contains\}$ be the set of atomic term predicates. An atomic term $a(M)$ in a Boolean tag type formula takes the set of morphological solutions M as input (free variable) and is of the form. $a(M) := \exists m \in M.m = \langle p, s, x, P, G, C \rangle . r_1 \circ r_2$ where $\circ \in \mathcal{O}$, $r_1 \in \{p, s, x, P, G, C\}$, $\exists A \in \mathcal{F}.r_1 \in A, r_2 \in A$. Informally, an atomic term indicates that a solution vector exists where a feature from the solution contains or exactly matches a constant value for the feature specified by the user.

MATAr *Boolean formulae* are of the following form.

- a is a MATAr formula where a is an atomic term.
- $\neg f$ is a MATAr formula where f is a Boolean formula. This is interpreted as the negation (complement) of words matching f.
- $f \vee g$ is a MATAr formula where f and g are Boolean MATAr formulae. This is interpreted as the disjunction (union) of words matching f with the words matching g.

Tag type: The set of tag types $\mathcal{T}$ contains tuples of the form $\langle l, f, d \rangle$ where l is a text label, f is a MATAr Boolean formula, and d is a visualization legend. The legend contains information such as the foreground and background colors, and the name, family, and size of the font.

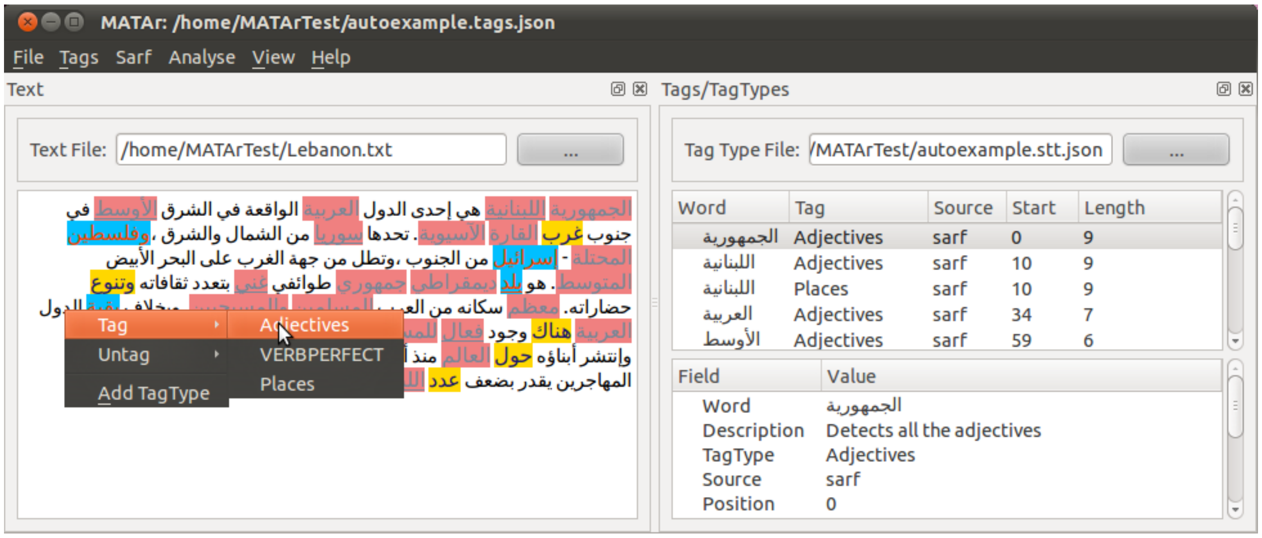
Evaluation: For each word $t_i \in T$, MATAr computes a Boolean value ($\{true, false\}$) for all atomic terms and Formulae. Then MATAr computes the set of tags $R \subseteq T \times \mathcal{T}$ such that $(t_i, tt_j) \in R$ iff the Boolean formula F_j associated with tag type tt_j is true for t_i.

Visualization: The visualization legends help the user visually distinguish the tag type and the class. Entity legends contain visualization information such as color and font. Relational entity legends contain information such as from and to arrows, edge labels, and frame lines.

IV. MATAr GUI

MATAr provides a user friendly interface to specify the atomic terms, the MATAr Boolean formulae, the tag types, and the legends. The MATAr GUI also allows the user to modify and correct the resulting tag set R and compute accuracy results that compare different tag sets. The accuracy results serve well as inter annotation agreement results when the tag sets come from two human annotators, or as evaluation results for learning and information extraction techniques.

The snapshot in Figure 1 shows the MATAr GUI with the tag type color sensitive view, the tag list view, and the tag description view. In the shown example, the user specified three tag types with labels “Adjectives”, “VERBPERFECT”, and “Places”. The “VERBPERFECT” tag type is based on a


MATAr: /home/MATArTest/autoexample.tags.json
File Tags Sarf Analyse View Help
Text
Tags/TagTypes
Tag
Adjectives
Untag
VERBPERFECT
Add TagType
Places

Text File: /home/MATArTest/Lebanon.txt

الجمهورية اللبنانية هي إحدى الدول العربية الواقعة في الشرق الأوسط في
جنوب غرب القارة الآسيوية. تحدها سوريا من الشمال والشرق، وفلسطين
المحتلة - إسرائيل من الجنوب، وتطل من جهة الغرب على البحر الأبيض
المتوسط. هو بلد ديمقراطي جمهوري طوائفي غني بتعدد ثقافاته وتنوع
حضاراته. معظم سكانه من العرب المسلمين والمسيحيين. وبخلاف بقية الدول
العربية هناك وجود فعال للمس
وإنتشر أبناؤه حول العالم منذ أ
المهاجرين يقدر بضعف عدد الل

Tag Type File: /MATArTest/autoexample.stt.json

Word	Tag	Source	Start	Length
الجمهورية	Adjectives	sarf	0	9
اللبنانية	Adjectives	sarf	10	9
اللبنانية	Places	sarf	10	9
العربية	Adjectives	sarf	34	7
الأوسط	Adjectives	sarf	59	6

Field	Value
Word	الجمهورية
Description	Detects all the adjectives
TagType	Adjectives
Source	sarf
Position	0

Fig. 1. MATAR main window with annotated text, tag descriptions, tag type legend properties, and manual tag edition menus.

simple formula that inspects the POS tag of the stem of the word. The "Adjectives" tag type is built on a formula that computes a disjunction between POS tags of the stem and the suffixes to include possessive forms as adjectives. The "Places" tag type is based on an abstract category specified by the user as a collection of stems and phrases.

The context sensitive menus in Figure 1 allow the user to manually change ore remove the tag of a selected word. The MATAR GUI also allows manual tag types that are not based on morphological features. These tags enable the users to build their own reference corpora without help from the morphological analyzer.

A. *Morphology based tag type editor*

The morphology based tag type editor shown in Figure 2 allows the user to write a tag type Boolean formula in a user friendly manner. The user first specifies atomic terms by selecting a feature from $\mathcal{F}$. The pattern is a regular expression that filters the feature values. This implicitly helps the user specify an *exact* versus a *partial* match of the desired morphological feature value.

Then the user can add the selected feature values to the atomic terms under the tag type name. The feature column has a context sensitive menu that allows negating the term. The value column has a context sensitive menu that can switch the operation between the values in $\mathcal{O} = \{isA, contains\}$. The right pane shows a description of the tag type and a set of legend descriptors.

For example, the user can select the feature *category* and associate it with the value *temporal unit*. Multiple feature/value pairs can be included in a single tag type definition with a disjunction semantics.

B. *Automated Tagging*

MATAR provides automatic tagging based on an Arabic morphological analyzer, *Sarf*. When the user triggers the automatic tagger, Sarf processes the text one word at a time. For each word, Sarf produces all valid solutions and returns the corresponding solution vectors. MATAR evaluates the Boolean formulae for each of the solutions and highlights the corresponding tags.

C. *Analysis*

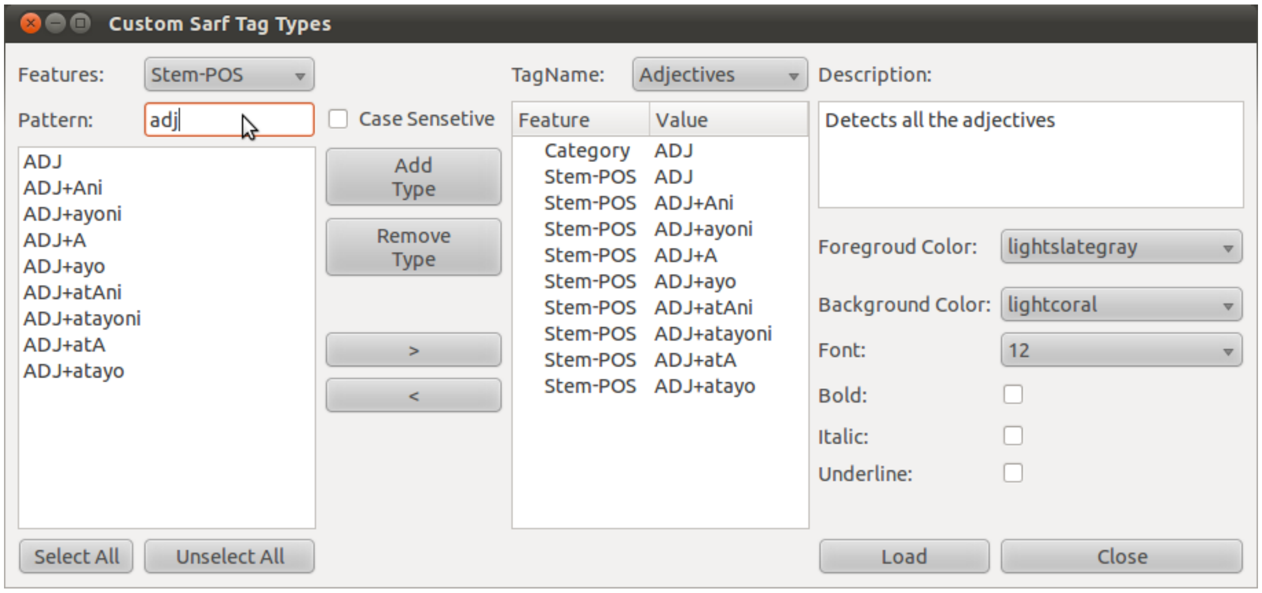
In addition to automatic and manual tagging, MATAR allows comparing tag sets and tag types applied to the same input text. MATAR comparator takes as input two tag sets R_1 and R_2 and two tag type sets $\mathcal{T}_1$ and $\mathcal{T}_2$. It produces a difference view for the tag types and a difference view for the tag sets. The tag type difference view shows the common tag types $\mathcal{T}_1 \cap \mathcal{T}_2$, the tag types in $\mathcal{T}_1$ and not in $\mathcal{T}_2$, and the tag types in $\mathcal{T}_2$ and not in $\mathcal{T}_1$.

Similarly, the tag set difference view shows $R_1 \cap R_2$, R_1/R_2 and R_2/R_1. The tag set difference view also shows the precision, recall and F-measure between the two sets. The metrics can be computed based on several predicates. The "Intersection" predicate returns true if a tag from R_1 intersects in text T with a tag in R_2. The "Exact" predicate returns true if a tag from R_1 exactly matches a tag in R_2. The "A includes B" predicate returns true if a tag from R_1 contains a tag from R_2. Finally, the "B includes A" predicate returns true if a tag from R_2 contains a tag from R_1.

In the difference view panes, the user can select a difference tag and accept it, or reject it to build a corrected corpora.

V. CONCLUSION

In this paper, we present MATAR, an open source morphology-based automatic tagger for Arabic. The tool enables the user to perform automatic and manual tagging. The user can define general purpose tag types or customized morphological tag types and tag chunks of text. The automatic tagging is based on Sarf, an Arabic morphological analyzer, and the resulting tags can be edited by the user. Moreover, the tool enables the user to compare two tag sets and returns statistical accuracy results including precision and recall measures. In the future, we plan to enhance the annotation capabilities



Custom Sarf Tag Types

Features: Stem-POS

Pattern: adj

Case Sensetive: ☐

ADJ
ADJ+Ani
ADJ+ayoni
ADJ+A
ADJ+ayo
ADJ+atAni
ADJ+atayoni
ADJ+atA
ADJ+atayo

TagName: Adjectives

Feature	Value
Category	ADJ
Stem-POS	ADJ
Stem-POS	ADJ+Ani
Stem-POS	ADJ+ayoni
Stem-POS	ADJ+A
Stem-POS	ADJ+ayo
Stem-POS	ADJ+atAni
Stem-POS	ADJ+atayoni
Stem-POS	ADJ+atA
Stem-POS	ADJ+atayo

Description: Detects all the adjectives

Foregroud Color: lightslategray

Background Color: lightcoral

Font: 12

Bold: ☐

Italic: ☐

Underline: ☐

Fig. 2. MATAR tag type Boolean formula editor.

of the tool by enabling the definition of rule based complex tag types along with actions performed on them.

References

[1] Imad A Al-Sughaiyer and Ibrahim A Al-Kharashi. Arabic morphological analysis techniques: A comprehensive survey. *Journal of the American Society for Information Science and Technology*, 55(3):189–213, 2003.

[2] Motasem Alrahabi, Amr Helmy Ibrahim, and Jean-Pierre Desclés. Semantic annotation of reported information in Arabic. *FLAIRS 2006, Floride, 11-13 Mai*, 2006.

[3] Mohamed Attia, M Rashwan, and MASAA Al-Badrashiny. Fassieh¯, a semi-automatic visual interactive tool for morphological, pos-tags, phonetic, and semantic annotation of Arabic text corpora. *Audio, Speech, and Language Processing, IEEE Transactions on*, 17(5):916–925, 2009.

[4] Bonnie J Dorr, Rebecca J Passonneau, David Farwell, Rebecca Green, Nizar Habash, Stephen Helmreich, Eduard Hovy, Lori Levin, Keith J Miller, Teruko Mitamura, et al. Interlingual annotation of parallel text corpora: a new framework for annotation and evaluation. *Natural Language Engineering*, 16(3):197, 2010.

[5] Kais Dukes, Eric Atwell, and Nizar Habash. Supervised collaboration for syntactic annotation of quranic Arabic. *Language Resources and Evaluation*, pages 1–30, 2011.

[6] Nizar Habash and Fatiha Sadat. Arabic preprocessing schemes for statistical machine translation. 2006.

[7] Hisham A Kholidy and N Chatterjee. Towards developing an Arabic word alignment annotation tool with some Arabic alignment guidelines. In *Intelligent Systems Design and Applications (ISDA), 2010 10th International Conference on*, pages 778–783. IEEE, 2010.

[8] Seth Kulick, Ann Bies, and Mohamed Maamouri. Consistent and flexible integration of morphological annotation in the arabic treebank. *Language Resources and Evaluation (LREC)*, 2010.

[9] Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. The penn arabic treebank: Building a large-scale annotated arabic corpus. In *NEMLAR Conference on Arabic Language Resources and Tools*, pages 102–109, 2004.

[10] Kazuaki Maeda and Stephanie Strassel. Annotation tools for large-scale corpus development: Using agtk at the linguistic data consortium. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation*, 2004.

[11] Jad Makhlouta, Fadi A. Zaraket, and Hamza Harkous. Arabic entity graph extraction using morphology, finite state machines, and graph transformations. In *Computational Linguistics and Intelligent Text Processing, CICLing*, pages 297–310, 2012.

[12] Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330, 1993.

[13] Thomas Morton and Jeremy LaCivita. Wordfreak: an open tool for linguistic annotation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: Demonstrations-Volume 4*, pages 17–18. Association for Computational Linguistics, 2003.

[14] Christoph Müller and Michael Strube. Multi-level annotation of linguistic data with MMAX2. *Corpus technology and language pedagogy: New resources, new tools, new methods*, 3:197–214, 2006.

[15] Layan M Bin Saleh and Hend S Al-Khalifa. AraTation: an Arabic semantic annotation tool. In *Proceedings of the 11th International Conference on Information Integration and Web-based Applications & Services*, pages 447–451. ACM, 2009.

[16] Otakar Smrz and Petr Pajas. Morphotrees of arabic and their annotation in the tred environment. In *Proceedings of the NEMLAR International Conference on Arabic Language Resources and Tools*, pages 38–41, 2004.

[17] Pontus Stenetorp, Sampo Pyysalo, Goran Topic, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. Brat: a web-based tool for nlp-assisted text annotation. *EACL 2012*, page 102, 2012.

[18] Naiwen Xue, Fei Xia, Fu-Dong Chiou, and Marta Palmer. The penn chinese treebank: Phrase structure annotation of a large corpus. *Natural Language Engineering*, 11(2):207, 2005.

[19] Fadi Zaraket and Jad Makhlouta. Arabic morphological analyzer with agglutinative affix morphemes and fusional concatenation rules. In *Proceedings of COLING 2012*, pages 517–526, Mumbai, India, December 2012.

[20] Fadi A. Zaraket and Jad Makhlouta. Arabic cross-document NLP for the hadith and biography literature. In *Florida Artificial Intelligence Research Society Conference (FLAIRS)*, Marco Island, Florida, May 2012.

[21] Fadi A. Zaraket and Jad Makhlouta. Arabic temporal entity extraction using morphological analysis. *International Journal of Computational Linguistics and Applications (IJCLA)*, 3:121–136, 2012.