Improving Domain Independent Question Parsing with Synthetic Treebanks

Halim-Antoine Boukaram, Nizar Habash,[†] Micheline Ziadee, and Majd Sakr[‡]
American University of Science and Technology, Lebanon
[†]New York University Abu Dhabi, UAE
[‡]Carnegie Mellon University, USA
{hboukaram,mziadee}@aust.edu.lb, nizar.habash@nyu.edu, msakr@cs.cmu.edu

Abstract

Automatic syntactic parsing for question constructions is a challenging task due to the paucity of training examples in most treebanks. The near absence of question constructions is due to the dominance of the news domain in treebanking efforts. In this paper, we compare two synthetic low-cost question treebank creation methods with a conventional manual high-cost annotation method in the context of three domains (news questions, political talk shows, and chatbots) for Modern Standard Arabic, a language with relatively low resources and rich morphology. Our results show that synthetic methods can be effective at significantly reducing parsing errors for a target domain without having to invest large resources on manual annotation; and the combination of manual and synthetic methods is our best domain-independent performer.

1 Introduction

Automatic syntactic parsing for questions is a challenging task since most treebanks are built from news domain articles that contain few questions (Hermjakob, 2001). Consequently, parsers can have difficulty with parsing questions (Hara et al., 2011; Gayo, 2011). This is a problem, especially in low resource languages like Arabic where the main gold standard treebank, the Penn Arabic TreeBank (PATB) (Maamouri et al., 2004), contains only 428 questions out of 12k annotated sentences (PATB Part 3).

In addition to the issue of sparse question data, parsing accuracy is further affected by the difference between the training data domain and the test data domain (Sekine, 1997; Haddow and Koehn, 2012; Van der Wees et al., 2015). The cost of manually building and labeling a treebank can be prohibitive; so different methods to maximize the impact of available human resources are utilized. One of these methods is model adaptation through using the output of a model trained on one domain (e.g., news) to annotate data from a different target domain (e.g., science fiction) (Su and Yan, 2017; Petrov et al., 2010). The automatically annotated data is then used to train a new model that has improved accuracy in the target domain. There are published efforts that deal specifically with adapting models for question parsing (Judge et al., 2006; Petrov et al., 2010; Seddah and Candito, 2016).

In this paper, we evaluate the efficacy of two low-cost synthetic treebank generation methods at improving the parsing accuracy of questions for Arabic in a number of domains. One of the methods we use relies on existing treebanks and uses phrase structure transformations to create questions from statements. The second method elicits generic unlexicalized templates from human users of a specific application. Another contribution of this paper is the creation of a manually annotated treebank of 988 questions (5.3k words) in two question-heavy domains to allow us to evaluate the three methods.

2 Methodology

Our basic approach is to train a syntactic parser with different combinations of treebanks (synthetic and manual) on top of a baseline of a commonly used treebank (PATB). We will evaluate the parser with a number of data sets from different domains with a high proportion of question constructions.

Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018), pages 214–221
Santa Fe, New Mexico, USA, August 25-26, 2018.

We created three treebanks using three different methods: one manually validated, and two synthetically constructed. The manual treebank covers two domains: talk shows and chatbots. The first synthetic question treebank is generated automatically from PATB using a number of manual question generation rules. The second is created from manual unlexicalized annotated question templates. Each synthetic corpus covers one domain of questions. We present next the different treebanks we created.

2.1 The Manual Treebank

We acquired the data used to build the manual treebank from two domains. This resulted in two subtreebanks, TalkShow and Chatbot, described in the next subsections. The raw data was automatically parsed using the Stanford parser (Klein and Manning, 2003; Green and Manning, 2010) with a model built from the PATB. The output (POS and syntactic structure) was validated by a human before being integrated into our manual treebank.

2.1.1 The TalkShow Treebank

We annotated 687 questions from the transcripts of a political TV talk show. After the initial automatic parsing, 54% of the questions were accepted without modification. We corrected 18% of the questions, and entirely re-annotated 28% of the questions when the automated process failed to produce reasonable results Although its domain is close to that of the PATB (news) there were some specific constructions in the TalkShow treebank which include: (a) Declarative questions; (b) Non-sentential utterances, e.g., 'Which state?'; (c) Questions where the question word is at the end of the phrase rather than the beginning, e.g.,'That state supports you, why?'; (d) Questions with vocative elements; and (e) Questions with topicalization, e.g., 'The problem, who caused it?'

2.1.2 The Chatbot Treebank

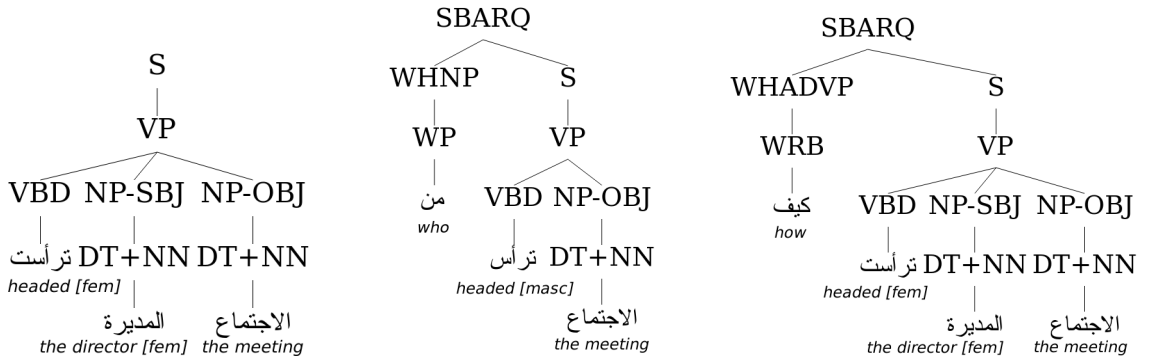
We annotated 162 questions from the CMUQ Hala logs. Hala (Makatchev et al., 2010), a roboceptionist that was deployed at the CMU Qatar campus, would answer users' typed questions in Arabic and English. These questions consisted of a high proportion of simple *where* questions (e.g., 'where is the bathroom?', due to user priming). In order to extend this conversational treebank, we ran a survey among Arabic speaking university students asking them what questions they have asked or would ask a university receptionist. This gave us an additional 139 questions for a total of 301 university-related questions. As with the TalkShow treebank, this treebank was annotated either by using the unmodified parser output (62% of questions), correcting the parser output (26%), or annotating manually (12%). Unlike TalkShow whose source is spoken language, Chatbot's source is written language.

2.2 The QGen$_{PATB}$ Synthetic Treebank

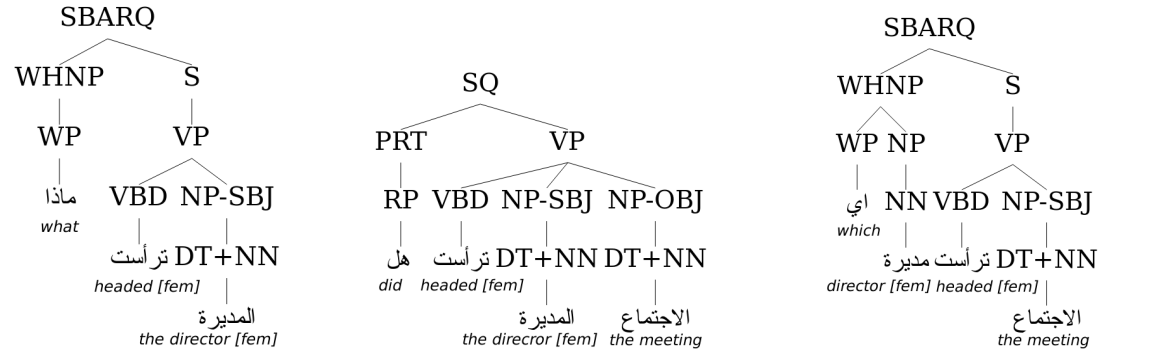
We decided to increase the number of annotated questions available to the model in order to increase its familiarity with interrogative structures. We implemented an automatic procedure to generate annotated questions similar to the latter stages of the work by Ali et al. (2010) and Heilman and Smith (2010) who used the parsed output of raw English data. Our procedure took as input an annotated PATB sentence, to which we added some extra gender, number, and rationality information (GNR) (Alkuhlani and Habash, 2011), and produced a set of annotated questions, following 21 question generators. Multiple question trees could be generated from each PATB tree. All questions began with a question word since this is the most common type of question.[1]

Each question-generating procedure worked by examining the semantic dash-tags available in the input PATB tree. The dash-tags used are: -SBJ (subject), -OBJ (object), -PRD (predicate), -DIR (direction), -LOC (locative), -TMP (temporal), and -CLR (closely related). The question generators implemented grammatical tree conversion rules on the input elements (Subject, Verb, etc.) to create questions with correct morphological agreement in terms of gender, number, and verb tense. Figure 1 shows some examples of automatically generated question structures.

[1] 93% of TalkShow questions and 100% of Chatbot questions begin with a question word (ignoring any sentence-initial conjunctions and interjections).



(a) The director headed the meeting. (b) Who headed the meeting? (c) How did the director head the meeting?



(d) What did the director head? (e) Did the director head the meeting? (f) Which director headed the meeting?

Figure 1: Example questions in QGen$_{PATB}$ (b,c,d,e,f) generated from original tree (a).

Some questions have sub variations to account for optional elements in the input sentence or the output question. For example, in Arabic, the subject can be inferred from the conjugation of the verb, so some pronouns can be added/removed from the sentence without affecting its meaning. Other variations take into consideration synonymous Arabic question words, e.g., ما *mA*[2] and ماذا *mAðA* 'what'; لما *limA* and لماذا *limAðA* 'why'; and أ Â and هل *hal* 'is it true that'.

Arabic has two types of sentences, verbal and nominal (Habash, 2010). Table 1 lists some question generators for the two types of sentences with optional elements encapsulated in square braces.

	Input Form	**Output Form**	**Input Sentence**		**Output Question**	
Verbal	Verb [Subj] Obj	Who Verb Obj	He took a flag	أخذ علما	Who took a flag ?	من أخذ علما؟
	Verb [Subj] Obj	What Verb [Subj]	He took a flag	أخذ علما	What did he take ?	ماذا أخذ؟
	Verb Obj Subj	Which Subj Verb Obj	Iran's ambassador attended the meeting	حضر اللّقاء سفير إيران	Which ambassador attended the meeting?	أي سفير حضر اللّقاء؟
	Verb [Subj] PP	Prep What Verb [Subj]	He is looking into the case	يبحث في القضية	Into what is he looking?	في ماذا يبحث؟
Nominal	Subj Pred	Who Pred	They (dual) are honest (dual)	انهما صادقان	Who is honest (singular)?	من صادق؟
	Subj Pred	What Pred	Its repercussions are negative	إنعكاساته سلبية	What is negative?	ماذا سلبي؟
	Subj Pred	Is Subj Pred	They are present	هم حاضرون	Are they present?	هل هم حاضرون؟
	Subj Pred	Is Subj Pred	He is outside	هو في الخارج	Is he outside?	أهو في الخارج؟

Table 1: Example questions of QGen$_{PATB}$.

The limitation of this synthetic technique is its dependence on an existing annotated treebank. However, its main advantage is that it can be easily reused to generate many question trees from any new non-question treebank in another domain with no added cost.

[2]Arabic script transliteration is presented in the Habash-Soudi-Buckwalter transliteration scheme (Habash et al., 2007).

2.3 The QTemp Synthetic Treebank

To address the lack of raw data in our desired domain (conversational), we tasked an in-house team with the generation of questions that they would ask a university receptionist. The tasks were divided into topics. The topics included *place questions* such as asking for directions, *process questions* such as obtaining academic transcripts, *event questions* such as graduations and expositions, and *degree questions* such as available majors. The questions were written as templates similar to Serban et al. (2016). Each question template (e.g., Where is %place%?) included a token (e.g., %place%) that could be replaced with real values (e.g., cafeteria, reception). Table 2 shows an example of an annotated question template and some generated questions for the place topic. In order to maximize coverage, we asked several team members to generate questions on the same topics.

Template: (SBARQ (WHADVP (WRB أين 'where')) (S (NP **%place%**)) (PUNC ?))
(SBARQ (WHADVP (WRB أين 'where')) (S (NP **(DT+NN** المدخل **'the entrance')**)) (PUNC ?))
(SBARQ (WHADVP (WRB أين 'where')) (S (NP **(DT+NNS** الصفوف **'the classrooms')**)) (PUNC ?))
(SBARQ (WHADVP (WRB أين 'where')) (S (NP **(NN** مكتب **'office') (NP (DT+NN** العميد **'the dean')**))(PUNC ?))

Table 2: Example of question generation for %place% question template.

Each question template was annotated by three in-house independent annotators with an average agreement rate of 92%. The differences were discussed and resolved resulting in a single annotation. When combining the annotated question templates with the annotated values of the tokens, care was taken to match for number, gender and case. Table 3 lists the number of templates and values annotated per topic. The number of generated questions per topic equals the number of question templates times the number of token values.

Topic	**Place**	**Process**	**Event**	**Degree**
# Templates	65	137	188	187
# Token Values	86	41	43	22

Table 3: Number of annotated templates and values.

One difference between QTemp and the PATB is the prevalence of first and second person singular verbs and pronouns in the former versus third person plural verbs and pronouns in the latter.

3 Experimental Setup

For all evaluations, the Stanford parser (Version 3.8.0) was used to train models for various combinations of train sets which were evaluated against all test sets. The parser's command-line *maxLength* argument was set to 30 so as to exclude longer sentences. All other configuration variables were left to default. Given the complexity of evaluating both segmentation and syntactic parsing together (Marton et al., 2013; Tsarfaty et al., 2012), we used the gold segmentation but the POS tags were predicted. Table 4 shows the distribution of treebanks between train and test sets, which are explained in the next subsections.

Treebank	**Domain**	**Train # Sentences (# Words)**	**Test # Sentences (# Words)**
ATB	News articles	10,836 (320,998)	794 (12,884)
ATBQ	News articles	N/A	67 (1,054)
TalkShow	Political talk show	544 (2,691)	143 (692)
Chatbot	Conversational	239 (1,505)	62 (441)
QGen$_{PATB}$	News articles (Synthetic)	962 (8,140)	N/A
QTemp	Conversational (Synthetic)	1,607 (13,099)	N/A

Table 4: The various treebanks used in terms of domain, training and testing sizes.

3.1 Train Sets

The Baseline The PATB is made up of 3 parts. Our baseline model was trained using the PATB (part 3) train dataset (PATB3.train) as defined by Diab et al. (2013).

The Manual Treebank Our manual treebank train set consisted of randomly selecting 80% of each of the TalkShow and Chatbot treebanks (783 questions). These two train sets were merged since their individual impact on their own test set is generally understood.

QGen$_{PATB}$ For QGen$_{PATB}$, 91k questions were generated from 10k trees in the PATB3.train set. Testing against the PATB3 development set (PATB3.dev), we found that using the entire QGen$_{PATB}$ treebank reduced accuracy due to the imbalance in question and non-question trees. All generated questions had the question word at the start of the phrase so the parser was biased against being able to detect the PATB questions that didn't begin with question words. This lead to an increase of SBARQs and SQs being tagged as SBARs and Ss respectively. To avoid the imbalance, the number of questions incorporated into the train set was empirically optimized by tuning against the PATB3.dev set, forming a random subset of QGen$_{PATB}$ with 962 questions.

QTemp QTemp consisted of a total of 23k questions. As with QGen$_{PATB}$, QTemp was tuned against PATB3.dev resulting in a random subset of 1.6k questions being used.

3.2 Test Sets

We report our results on four test sets. First is the standard PATB3 test set (Diab et al., 2013) which has only a small proportion of questions. Second is a question rich test set (PATBQ) created by merging trees with question structures (SBARQ, SQ) from PATB (parts 1 and 2), and PATB3.test (Diab et al., 2013). Third and fourth are the 20% (non-train) of the TalkShow and Chatbot treebanks, respectively. We obviously do not report on test portions from the synthetic treebanks because they are not manually validated.

4 Results

The parsing scores for different combinations (C1-C6) of train sets (shown in the second column) are shown in Table 5. The *Average Q* row is a macro average of the three question test sets (PATBQ, TalkShow, and Chatbot).

	Corpus	Baseline	C1	C2	C3	C4	C5	C6	All	CQ	Error Reduction All over Baseline	Error Reduction Synthetic (C5) over Baseline
Train	**PATB**	✓	✓	✓	✓	✓	✓	✓	✓			
	QGen$_{PATB}$ (Synthetic)		✓		✓		✓		✓	✓		
	QTemp (Synthetic)					✓	✓	✓	✓	✓		
	TalkShow+Chatbot			✓	✓			✓	✓	✓		
Test	**PATB**	80.6	80.6	80.6	80.7	80.7	80.6	80.8	**80.9**	43.0	1%	1.2%
	PATBQ	73.8	74.1	74.9	74.9	73.8	74.0	75.8	**75.9**	53.1	8%	4.1%
	TalkShow	88.2	88.2	91.4	92.1	87.5	87.3	92.7	**92.9**	83.2	38%	-8.2%
	Chatbot	90.5	90.7	93.3	93.0	93.6	93.6	**94.1**	**94.1**	86.7	40%	32.3%
	Average Q	84.2	84.3	86.5	86.7	85.0	84.9	87.5	**87.6**	74.3	22%	4.9%

Table 5: F-scores for Baseline to All models. C1-C6 and CQ represent the different combination of training sets which are shown in the second column.

The TalkShow+Chatbot treebank data was the best single addition to the baseline (C2 in Table 5), but is also the most costly and most difficult to build. In experiments not shown here, we found that excluding Chatbot or TalkShow from C2 reduces the performance on the TalkShow and Chatbot tests, respectively, which clearly shows that the two training sets boosted each other's test set.

The positive impact of the synthetic QTemp (C4) train set on the Chatbot test set was similar to that of the TalkShow+Chatbot train set (C2). However, QTemp had a negative impact on the TalkShow test

set due to the domain specific properties mentioned in 2.1.1. This suggests that synthetic data (if used alone) is beneficial for in-domain cases; but may have a negligible to negative effect on out-of-domain tests. For similar reasons, QGen$_{PATB}$ (C1) primarily had a positive impact on the PATBQ test set.

The cumulative model (All in Table 5), which includes all manual and synthetic train sets, was the most accurate suggesting added value for a combination of techniques. However, The model trained with only the question train sets (CQ in Table 5), suffered due to reduced coverage thereby justifying why PATB should always be part of the training data.

We used a two-tailed paired t-Test to evaluate the significance of the error reduction exhibited by the cumulative model over the baseline. The p-values were 0.0039 for PATB, 0.086 for PATBQ, 8×10^{-5} for TalkShow, and 0.031 for Chatbot. Only the results of the PATBQ were not significant due to the small test size and the relatively small error reduction.

5 Related Work

There is a large body of literature on automatic parsing for English and other languages (Charniak, 1997; Steedman et al., 2003; Judge et al., 2006; Kübler et al., 2009; Green and Manning, 2010; Petrov et al., 2010; Zaki et al., 2016). Some of these efforts dealt with the issue of automatic training data enrichment to boost parsing accuracy. Steedman et al. (2003) showed that self training, i.e., using the output of a parser on raw text as additional training data, did not do as well as co-training, i.e., iteratively retraining two (or more) parsers on each other's output. We do not report here on self training experiments that we did because they gave negative results. Petrov et al. (2010) showed that training with the output of a different parser can increase accuracy. In this paper, we only worked with one parser.

Our work is more closely related to the work by Judge et al. (2006) on the English Question Bank. They obtained a 51.7% reduction in the error rate of parsing questions by adding manually annotated questions to their train set. In our case, using only the synthetic QTemp treebank, we achieved a 32.3% on the Chatbot test set. Also similar to Judge et al. (2006), was the result of training only on the question sets which gave reduced parsing accuracy across all test sets (particularly the statement-rich test set).

In regards to work on Arabic parsing, we use similar but not directly comparable data sets and parameters to Green and Manning (2010). Our baseline is similar to what they achieved. There is a growing body of research on Arabic dependency parsing (Habash and Roth, 2009; Marton et al., 2013; Taji et al., 2017; Taji et al., 2018). Most recently, Taji et al. (2018) introduced a travel domain treebank that had a very high proportion of question constructions (40% of the trees).

The annotation speeds reported by Habash and Roth (2009) were between 540 and 715 tokens/hour. They also reported annotation speeds of 250-300 tokens/hour for the PATB and around 75 tokens/hour for the Prague Arabic Dependancy Treebank (PADT) (Smrž and Hajic, 2006). Our annotation speeds for QTemp are not directly comparable given the difference in the tasks, but they were around 4.2 templates/hour which included the time taken for the three annotators to discuss any differences.

6 Conclusion and Future Work

In this paper, we have shown that low-cost methods to produce synthetic domain-specific training data can greatly improve syntactic parsing accuracy for in-domain tests. Using synthetic training data, we achieved an error reduction rate that was comparable to that achieved with manually annotated data (in-domain). However, the synthetic data was found to have a negligible to negative effect on out-of-domain tests. The resulting synthetic and manual data is useful for other researchers working in Arabic computational linguistics especially with regards to analyzing questions.

In the future, we plan to write additional question-generating procedures targeting particular question constructions such as 'how much/many' and 'which' that are sparse in the current QGen$_{PATB}$. Inspired by Petrov et al. (2010), we plan to evaluate other parsers against our test sets to identify which parser would be best suited to provide automatic training data. We will also test whether the synthetic approach is generalizable to other languages. Finally, we will investigate how the improvement in syntactic parsing affects a question answering task and compare that against a pure grammar rule-based approach.

References

Husam Ali, Yllias Chali, and Sadid A Hasan. 2010. Automation of question generation from sentences. In *Proceedings of QG2010: The Third Workshop on Question Generation*, pages 58–67.

Sarah Alkuhlani and Nizar Habash. 2011. A corpus for modeling morpho-syntactic agreement in Arabic: gender, number and rationality. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 357–362. Association for Computational Linguistics.

Eugene Charniak. 1997. Statistical parsing with a context-free grammar and word statistics. *AAAI/IAAI*, 2005(598-603):18.

Mona Diab, Nizar Habash, Owen Rambow, and Ryan Roth. 2013. LDC Arabic treebanks and associated corpora: Data divisions manual. *arXiv preprint arXiv:1309.5652*.

Iria Gayo. 2011. Question parsing for QA in Spanish. In *Proceedings of the Second Student Research Workshop associated with RANLP 2011*, pages 73–78.

Spence Green and Christopher D Manning. 2010. Better Arabic parsing: Baselines, evaluations, and analysis. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 394–402. Association for Computational Linguistics.

Nizar Habash and Ryan M Roth. 2009. CATiB: The Columbia Arabic treebank. In *Proceedings of the ACL-IJCNLP 2009 conference short papers*, pages 221–224. Association for Computational Linguistics.

Nizar Habash, Abdelhadi Soudi, and Tim Buckwalter. 2007. On Arabic Transliteration. In A. van den Bosch and A. Soudi, editors, *Arabic Computational Morphology: Knowledge-based and Empirical Methods*. Springer.

Nizar Y Habash. 2010. Introduction to Arabic natural language processing. *Synthesis Lectures on Human Language Technologies*, 3(1):1–187.

Barry Haddow and Philipp Koehn. 2012. Analysing the effect of out-of-domain data on SMT systems. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 422–432. Association for Computational Linguistics.

Tadayoshi Hara, Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2011. Exploring difficulties in parsing imperatives and questions. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 749–757.

Michael Heilman and Noah A Smith. 2010. Good question! statistical ranking for question generation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 609–617. Association for Computational Linguistics.

Ulf Hermjakob. 2001. Parsing and question classification for question answering. In *Proceedings of the workshop on Open-domain question answering-Volume 12*, pages 1–6. Association for Computational Linguistics.

John Judge, Aoife Cahill, and Josef Van Genabith. 2006. Questionbank: Creating a corpus of parse-annotated questions. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 497–504. Association for Computational Linguistics.

Dan Klein and Christopher D Manning. 2003. Fast exact inference with a factored model for natural language parsing. In *Advances in neural information processing systems*, pages 3–10.

Sandra Kübler, Ryan McDonald, and Joakim Nivre. 2009. Dependency parsing. *Synthesis Lectures on Human Language Technologies*, 1(1):1–127.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. 2004. The Penn Arabic Treebank: Building a large-scale annotated Arabic corpus. In *NEMLAR conference on Arabic language resources and tools*, volume 27, pages 466–467. Cairo.

Maxim Makatchev, Imran Fanaswala, Ameer Abdulsalam, Brett Browning, Wael Ghazzawi, Majd Sakr, and Reid Simmons. 2010. Dialogue patterns of an Arabic robot receptionist. In *Human-Robot Interaction (HRI), 2010 5th ACM/IEEE International Conference on*, pages 167–168. IEEE.

Yuval Marton, Nizar Habash, and Owen Rambow. 2013. Dependency parsing of modern standard Arabic with lexical and inflectional features. *Computational Linguistics*, 39(1):161–194.

Slav Petrov, Pi-Chuan Chang, Michael Ringgaard, and Hiyan Alshawi. 2010. Uptraining for accurate deterministic question parsing. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 705–713, Stroudsburg, PA, USA. Association for Computational Linguistics.

Djamé Seddah and Marie Candito. 2016. Hard time parsing questions: Building a questionbank for French. In *Tenth International Conference on Language Resources and Evaluation (LREC 2016)*.

Satoshi Sekine. 1997. The domain dependence of parsing. ANLC '97, pages 96–102, Stroudsburg, PA, USA. Association for Computational Linguistics.

Iulian Vlad Serban, Alberto García-Durán, Caglar Gulcehre, Sungjin Ahn, Sarath Chandar, Aaron Courville, and Yoshua Bengio. 2016. Generating factoid questions with recurrent neural networks: The 30m factoid question-answer corpus. *arXiv preprint arXiv:1603.06807*.

Otakar Smrž and Jan Hajic. 2006. The other Arabic treebank: Prague dependencies and functions. *Arabic computational linguistics: Current implementations. CSLI Publications*, 104.

Mark Steedman, Miles Osborne, Anoop Sarkar, Stephen Clark, Rebecca Hwa, Julia Hockenmaier, Paul Ruhlen, Steven Baker, and Jeremiah Crim. 2003. Bootstrapping statistical parsers from small datasets. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 1*, pages 331–338. Association for Computational Linguistics.

Yu Su and Xifeng Yan. 2017. Cross-domain semantic parsing via paraphrasing. *CoRR*, abs/1704.05974.

Dima Taji, Nizar Habash, and Daniel Zeman. 2017. Universal dependencies for Arabic. In *Proceedings of the Third Arabic Natural Language Processing Workshop*, pages 166–176.

Dima Taji, Jamila El Gizuli, and Nizar Habash. 2018. An Arabic dependency treebank in the travel domain. In *Proceedings of the 3rd Workshop on Open-Source Arabic Corpora and Processing Tools*, Miyazaki, Japan.

Reut Tsarfaty, Joakim Nivre, and Evelina Andersson. 2012. Joint evaluation of morphological segmentation and syntactic parsing. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 6–10. Association for Computational Linguistics.

Marlies Van der Wees, Arianna Bisazza, Wouter Weerkamp, and Christof Monz. 2015. What's in a domain? analyzing genre and topic differences in statistical machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, pages 560–566.

Youssef Zaki, Haisam Hajjar, Mohamad Hajjar, and Gilles Bernard. 2016. A survey of syntactic parsers of Arabic language. In *Proceedings of the International Conference on Big Data and Advanced Wireless Technologies*, page 31. ACM.