Web Data Harvesting for Speech Understanding Grammar Induction

Ioannis Klasinas, Alexandros Potamianos, Elias Iosif, Spiros Georgiladakis, Gianluca Mameli[†]

Dept. of Electronic & Computer Engineering, Tech. Univ. of Crete, Chania, Greece
[†]Expert System, Trento, Italy
iklasinas@isc.tuc.gr, {potam,iosife}@telecom.tuc.gr, spgeo@intelligence.tuc.gr

Abstract

The development of a grammar for a spoken dialogue system can be greatly accelerated by using a corpus describing the application. However the development of such a corpus is a slow and expensive process. This paper proposes unsupervised methods for finding relevant corpora in the Web and mining the most informative parts. We show that by utilizing perplexity we are able to increase the in-domainess (precision) of the mined corpora, while by utilizing the rank of the web search engine we can increase the generalizability (recall). The results show that using only unsupervised and language independent methods we can compete with corpora created manually with expert knowledge.

Index Terms: spoken dialog systems, grammar induction, speech understanding, web harvesting, language modeling.

1. Introduction

In the last two decades, interactive spoken dialogue systems (SDS) have moved from research prototypes to real-life commercial applications. Still, one major roadblock in SDS prototyping is that they are not easily and quickly portable to new domains or languages. The design and implementation of speech understanding grammars (both statistical and finite-state) requires significant expertise and in-domain data. Specifically, SDS language engineers typically start from a bootstrap grammar and augment it manually with new fragments extracted from corpora and other resources. Alternatively the grammars can be induced automatically or semi-automatically (i.e., machine-assisted) from a corpus using a relevant set of algorithms and tools [1, 2]. Manual methods result in grammars that typically achieve high parsing precision and poor recall. Automatic methods are faster, cheaper and produce grammars with good coverage, but they tend to overgeneralize (poor precision). Both manual and automatic methods require a corpus; in fact, the performance of the resulting grammars is very much affected by the quality of the corpus. Just like the grammar, the quality of a corpus can be judged in terms of the generalized notions of precision and recall, namely, to what degree is the corpus in-domain (precision) and how rich is the corpus in terms of lexical, syntactic and semantic variability (recall).

In this paper, we investigate how starting for a bootstrap grammar (and optionally a small in-domain corpus), a spoken dialogue system developer can 1) harvest data from the web using queries formulated from the bootstrap grammar, 2) filter and select a relevant and linguistically rich subset of these data and 3) perform grammar induction on the corpus in order to enhance the bootstrap grammar. The process is evaluated both in term of harvested corpus in-domainess and linguistic richness, as well as grammar induction algorithm performance (precision and recall). The main contributions of this work are the introduction of: 1) domain/pragmatic constraints for corpus selection that are automatically induced from the corpus and significantly improve grammar induction performance, 2) a perplexity ratio metric for enhancing in-domainess of the selected sub-corpus (that does away for the need of an in-domain corpus for filtering) and 3) a detailed end-to-end evaluation of the algorithms for speech understanding grammar induction (vs speech recognition grammars that are usually investigated in the literature).

The remainder of this paper is organized as follows. First with an overview of methods proposed for grammar induction from a corpus is presented, along with a short description of our method in Section 2. In Section 3, a summary of the state-of-the-art in web data harvesting is presented followed by our method for query generation from a bootstrap grammar. We then proceed to discuss the data filtering/selection process and the resulting corpora in Sections 4 and 5. We conclude with the evaluation of the results and the discussion of the findings.

2. Grammar Induction from a Corpus

Data-driven approaches to grammar induction rely solely on corpora (bottom-up) of transcribed utterances [1, 2]. Such corpora serve as in-domain data and the success of data-driven approaches is strongly depended on the availability of them. The main idea here is to exploit the linguistic environment (context) for the identification of lexical and syntactic regularities that imply semantic similarity. The first step for inducing grammars according to the bottom-up paradigm is the induction of semantic classes (corresponding to terminal concepts in the domain taxonomy) based on the well-established distributional hypothesis of meaning stating that "similarity of context implies similarity of meaning" [3]. Contextual similarity can be estimated according to a variety of metrics, [4], that compute the divergence of n-gram distributions, e.g., Kullback-Leibler divergence. The typical algorithm of bottom-up grammar creation operates iteratively [1, 2]. At each iteration a number of chunks and semantic classes are automatically generated, while the corpus is appropriately updated (chunks and members of the same semantic class are substituted by an artificial label). In essence, this is an agglomerative clustering algorithm. Variations of the aforementioned algorithm include combination of similarity metrics, [5, 6], and soft-clustering [7] for the creation of semantic classes. The data-driven paradigm of grammar creation (induction) requires a considerable amount of data that is rarely available during the system design/development phase for a new domain [8]. The main drawback of data-driven approaches is the problem of data sparseness, which may affect the coverage of the grammar. Further, data-driven approaches induce syntactic grammars but do not learn their correspond-

ing meanings, for this purpose an additional step is required of parsing the grammar fragments and attaching them to the domain ontology [9]. Also, in many cases it was observed that the fully automated bottom-up paradigm results to grammars of moderate quality [10], especially on corpora containing longer sentences and more lexical variety [11]. Finally, algorithms focusing on cross-lingual grammar induction, like CLIoS [12], are often even more resource-intensive, as they require training corpora of parallel text and sometimes also a grammar for one of the languages.

2.1. Our Approach

Given a corpus, the overall process of grammar induction consists of the following main steps: 1) named entity recognition, e.g., "New York", 2) induction of semantic classes (concepts) of terminals, e.g., <CITY>=("New York", "Boston", ...), 3) extraction of grammar fragments (chunks), e.g., "departing <CITY>" and "leaving <CITY>", and 4) induction of grammar rules, e.g., <DEP_CITY> = ("departing" | "leaving") <CITY>. A fully-automatic, unsupervised scheme of this process can be implemented as a pipeline of the aforementioned steps, where each step depends on the previous one. For example, the extraction of grammar fragments builds upon the availability of semantic classes. A semi-automatic, supervised scheme can be also implemented in which feedback from a grammar developer is enabled at severals phases of the process, e.g., manual correction of the induced semantic classes.

In this work, we focus on the task of unsupervised induction of semantic classes in order to evaluate the several approaches for corpora creation from web-harvested data. The basic idea is the automatic creation of clusters that include semantically similar terminals. Our approach relies on the distributional hypothesis of meaning, i.e., "similarity of context implies similarity of meaning" [3]. At the sentential level a word (terminal) w is considered with its neighboring words in the left and right contexts: $w_1^L \ w \ w_1^R$. The semantic distance between two words, w_x and w_y, is estimated as the *Manhattan-norm* (M) of their respective bigram probability distributions of left and right contexts [2]. For example, the left-context M is defined as:

$$M^L(w_x, w_y) = \sum_{i=1}^{N} \mid p(w_i^L|w_x) - p(w_i^L|w_y) \mid, \qquad (1)$$

where $V = (w_1, w_2, ..., w_N)$ is the corpus vocabulary. Also, note that $M^L(w_x, w_y) \equiv M^L(w_y, w_x)$. The semantic distance between w_x and w_y is estimated as the sum of the left- and right-context M, i.e., $M(w_x, w_y) = M^L(w_x, w_y) + M^R(w_x, w_y)$. The induction of semantic classes is performed by applying a clustering algorithm, which is fed with the pairwise semantic distances of the words of interest.

3. Data Harvesting

A popular solution to the data sparseness bottleneck is to harvest in-domain data from the web. Recently, this has been an active research area both for SDS systems and language modeling in general. Data harvesting is performed in two steps: (i) query formulation, and (ii) selection of relevant documents or sentences. Posing the appropriate queries is important both for obtaining in-domain and linguistically diverse sentences. In [13], an in-domain language model was used to identify the most appropriate n-grams to use as web queries. A more sophisticated query formulation was proposed in [9], where from each in-domain utterance a set of queries of varying length and complexity was generated. These approaches assume the availability of in-domain data (even if limited) for the successful formulation of queries; this dependency is also not eliminated when using a lexicalized domain ontology to formulate the queries, as in [14]. Selecting the most relevant sentences that get returned from web queries is typically done using statistical similarity metrics between in domain data and retrieved documents, for example the BLEU metric [15] of n-gram similarity in [9] and a metric of relative entropy (Kullback-Leibler) in [13]. In cases where in-domain data is not available, cf. [14], heuristics (pronouns, sentence length, wh-questions) and matches with out-of-domain language models can be used to identify sentences for training SDS grammars. In [9], the produced grammar fragments are also parsed and attached to the domain ontology. Harvesting web data can produce high-quality grammars while requiring up to 10 times less in-domain data [9].

3.1. Web query generation from grammar fragments

In a typical speech understanding grammar development cycle, the developer starts from user requirements (often expressed as request types or a small corpus) and then encodes this information in hand-crafted grammar. Our goal is starting from this limited-coverage bootstrap grammar to harvest a corpus using web queries, in order to enhance the grammar via rule induction from this corpus. Our approach is to generate queries from phrases found in the bootstrap grammar. As far as we know, this is the first time that queries are generated from a grammar, although, the method is similar to [9] where n-gram fragments can be extracted from an already available corpus.

Starting from an existing grammar is a more realistic scenario for spoken dialogue system development, as well as, presents the advantage that specific parts of the grammar (concepts) can be enhanced as requested by the developer. If the grammar is small, it might be possible to generate all the phrases described and feed them to the web search engine. Usually, the size of the grammar prohibits such an exhaustive search. Instead, fragments from the grammar itself are created ignoring instantiations of terminal concepts that would increase the complexity too much. For example, consider the following rule present in the initial grammar; ***DEPT.CITY*** —> [*DEPART* | *DEPARTING* | *LEAVE* | *LEAVING* | *LEFT*] (*FROM* | *BETWEEN* | *OUT OF*) ***CITY***. In this grammar ***CITY*** can be replaced with 2000 city names. As result, the above rule would create 48000 phrases. To overcome this problem terminal concept ***CITY*** instantiations are ignored, resulting in just 18 queries created for the above case. To further narrow down the results domain (or pragmatic) constraints are appended to each query that describe the application domain. We believe that this does not pose a big problem to the applicability of the method to different domains/languages, since minimal human intervention is required. Example queries for the travel domain are presented next:

"DEPARTED BETWEEN" AND (FLIGHT OR TRAVEL OR AIRPORT)
"DEPARTED FROM" AND (FLIGHT OR TRAVEL OR AIRPORT)
"DEPARTED OUT OF" AND (FLIGHT OR TRAVEL OR AIRPORT)

The method described above could also use an ontology instead of a grammar, if available, as discussed in [14].

4. Corpus selection and filtering

We investigate two criteria for selecting a subset of the downloaded corpus namely: perplexity and pragmatic constraints.

4.1. Perplexity

The perplexity of a sentence W_1^I of length I according to a probability model P is defined as $PPL_P(W_1^I) = 10^{-\log P(W_1^I)/I}$. High probability for a given sentence implies that this sentence is similar to the distribution of the model, leading to low perplexity. Perplexity is a popular criterion [16, 17, 18] for selecting corpora for n-gram language model training. In this paper, we show that in the absence of an in-domain corpus one can use the downloaded corpus as the model P to select a low perplexity subset of the corpus for speech understanding grammar induction.

4.2. Defining pragmatic constraints

Pragmatic constraints, i.e., words that have high application domain saliency, can be used in the filtering step, to pick the most informative sentences from the downloaded corpus. Instead of manually selecting such words, we propose to find this set of constraints in an unsupervised way. Generally speaking, highly salient domain words would appear much more frequently in an in-domain (foreground) corpus rather than in a general-purpose (background) corpus. In addition, pragmatic constraints should appear in the majority of the in-domain corpus documents, i.e., will be evenly spread in the foreground corpus. If $P_{for}(w)$ is the probability of a word according to the foreground model and $P_{bck}(w)$ according to the background model, then their ratio multiplied by the percent of in-domain documents that contain this word $D(w)$ can provide a good criterion for selecting salient words, i.e., $P_I(w) = D(w)\ P_{for}(w)/P_{bck}(w)$. If an in-domain corpus is not available, the downloaded corpus is used instead. Computing this metric for the vocabulary of the downloaded corpus the most informative words can be selected.

5. Corpora and Experimental Procedure

The experiments focus on an English travel domain grammar. The grammar consists of *terminal concepts* like <**cityname**> which form concept fragments with lexical only right hand, like in <**cityname**> $\rightarrow$ **NEY YORK**. Higher level rules are called *grammar rules* like <**TOCITY**> which form grammar fragments like <**ARR.CITY**> $\rightarrow$ **TO** <**cityname**>. The gold standard grammar statistics are detailed in Table 1.

terminals	term. instanc.	grammar rules	gram. fragm.
83	3299	47	2020

Table 1: Golden standard grammar.

In addition to the automatically-generated web harvested corpora we also use four dialogue corpora (for comparison purposes) collected from various scenarios: question-answering (ATIS corpus of 1560 sentences), human-human dialogue between travel agent and customers (1910 sentences), human-system data collected from a DARPA Communicator system (11516 utterances) and data collected during system development using a Wizard of Oz (WoZ) scenario (1295 sentences). A fifth corpus is collected from the web using crawling and post-filtering using hand-crafted rules. Starting from the Q&A corpus a number of representative queries are created by an expert, that describe well the characteristics of the corpus. Using a web search engine, relevant documents are fetched, and the created corpus is filtered using hand crafted rules that select conversational sentences relevant to the domain.

The automatically created corpus was harvested from a set of 1832 queries that were created from a subset of the initial english grammar (50% of the grammar rules were used in this process). The pragmatic constraints used were *FLIGHT*, *TRAVEL*, or *AIRPORT*. Using the Yahoo! web search engine the top 250 documents were downloaded for each query Each webpage was stripped off the html code using the freely available Javaparser and then split into sentences using [19]. After this preprocessing the corpus available totalled 19M lines. Since the focus is on conversational corpora, only sentences with a length between 5 and 50 words were kept. Then a 4-gram language model was trained on out of domain english corpus (News+EuroParl) and the downloaded sentences were ranked according to perplexity. The motivation was to filter out sentences that contain spelling errors and non (very) natural language. A threshold of 6000 for perplexity was experimentally defined to be a good tradeoff.

The corpora downloaded was filtered in a number of ways. First sentences were randomly selected from the corpus (*Random*). To explore the importance of the ranking of the web search engine corpora were created from the first hit only (*Top 1*), as well as slices of 50 hits (*Rank 0-50*). For perplexity-based corpus filtering, a fourgram language model was trained on the downloaded corpus[1]. The downloaded sentences were ranked according to perplexity and subcorpora for various perplexity ranges were evaluated. Here we report results for sentences with a perplexity lower than 15 for the web corpus (*WebPPL<15*). We also report results for filtering using the Q&A corpus as the in-domain corpus with perplexity lower than 250 (*QAPPL<250*). Finally the corpora were filtered according to a set of pragmatic constraints. The top 3 words estimated automatically from the criterion in Section 4.2 were used from the list: FLIGHT, AIRPORT, TRAVEL, YOUR, YOU, AIRLINES, HOTEL, CHECK, CHEAP, MAP, BOOKING, FLYING, AIR etc. Constraints were also applied in conjunction with the filters described above, e.g., *QAPPL<250+constraints*, *Top1+constraints*.

5.1. Terminal Concept Induction

For the estimation of probabilities used in (1) defined in Section 2.1, a bigram language model was built using the CMU Statistical Language Modeling toolkit [20], applying Witten-Bell discounting and using back-off weights to compute the probabilities of unseen bigrams. The computation of pairwise semantic distances was focused only to the terminals that appear within the domain vocabulary, i.e., terminals of ground-truth. The induction of semantic classes was performed by applying agglomerative clustering using the CLUTO toolkit [21]. We experimented with number of clusters ranging from 25 to 400.

6. Experimental Results

The evaluation of the corpora is performed at two levels: 1) corpus analysis using the gold standard grammar and the parser to investigate the in-domainess and richness of the corpus (percent of sentences with a partial parse, total/unique number of terminal concept instances and grammar fragments) and 2) performance of the grammar induction algorithm in terms of weighted precision, recall and F-measure.

[1] Alternatively one can train a language model on the in-domain dialogue corpora (if available). We report results for both cases.

Corpus	Words	Sentences Partially Parsed (percent)	Terminal Instances (per word.)	Grammar Fragments (per word)	Terminal Instances (unique)	Terminal Concepts (unique)	Grammar Fragments (unique)	Grammar Rules (unique)
Q&A	18052	**99.6**	**0.33**	**0.46**	330	53	199	36
WoZ	9926	85.4	0.30	0.25	406	64	173	38
Human-Human	21432	87.0	0.28	0.26	374	65	221	39
Human-System	75182	86.7	0.32	0.30	811	78	307	44
Manually harvested	216057	94.1	0.19	0.18	**1019**	**83**	**356**	**46**
Random	266704	59.7	0.10	0.06	1102	**82**	196	40
Top 1	267413	70.8	0.12	0.09	1382	81	245	**44**
Rank 0-50	259697	60.9	0.11	0.07	1189	79	200	41
WebPPL < 15	319476	76.4	0.13	0.10	1514	77	212	42
Top1+constraints	299961	93.5	0.15	0.13	**1549**	79	**268**	43
WebPPL+constraints	278514	**95.2**	**0.16**	**0.14**	1490	72	191	40
QAPPL < 250	93952	70.7	0.25	0.34	**925**	**66**	**203**	34
QAPPL+constraints	40659	**98.3**	**0.29**	**0.42**	702	51	156	**36**

Table 2: In-domain and richness metrics for each corpus using the gold standard grammar.

Corpus	Prec.	Recal	F-meas.	Term. Conc.	Term. Instan.
Q&A	**0.52**	0.40	**0.45**	14	144
WoZ	0.41	0.33	0.37	14	128
Human-Human	0.42	0.32	0.36	15	110
Human-System	0.41	0.34	0.37	24	255
Manually harvested	0.46	**0.41**	0.43	**26**	**399**
Random	0.42	0.30	0.35	**27**	313
Top 1 doc	0.40	0.29	0.34	22	371
Rank 0-50	0.42	0.30	0.35	22	326
WebPPL$<$15	0.39	0.32	0.35	10	452
Top1+constraints	0.38	0.35	0.36	15	497
WebPPL$<$15+constrain.	**0.49**	**0.45**	**0.47**	16	**638**
QAPPL$<$250	0.49	0.40	0.44	**16**	**365**
QAPPL$<$250+constrain.	**0.56**	**0.45**	**0.50**	13	312

Table 3: Grammar induction evaluation on different corpora.

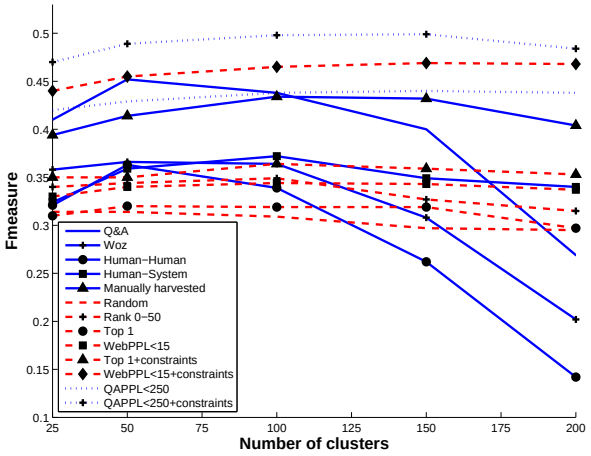

Figure 1: F-measure for various corpora and number of clusters.

6.1. Corpus analysis: in-domain and richness metrics

Using the parser provides a straightforward way to measure the number of sentences that contain relevant (in-domain concepts), as well as the total number of grammar rules that are exercised by the corpus (richness). The results are shown in Table 2 in terms of total and unique grammar fragments and terminal concepts present, as well as percent of sentences (partially) parsed by the grammar. From the manually created corpora, the *Q&A* corpus is the most relevant (in-domain) one; 33% of the words are instances of terminal concepts and 46% of the words are grammar fragments. However this corpus only covers a small part of the grammar. The *manually harvested* corpus includes many more rule instances and is the richer out of the five manually created corpora. Among the automatically harvested corpora, using perplexity as a filtering criterion leads to a corpus that is more relevant but also less rich than the original corpus. Using the web corpus proper for perplexity estimation provides a good trade-off between in-domainess and richness. Adding pragmatic constraints significantly improves the relevance of the corpus with a small loss in generalizabiity. Overall, combining perplexity with pragmatic constraints provides best results.

6.2. Evaluation of terminal class induction

For grammar induction evaluation, the grammar rules that include only terminals were used as ground-truth. Each induced class was mapped to the corresponding (best match) ground-truth rule. Class-weighted precision, recall, and F-measure were used as evaluation metrics. The grammar induction results are given in Table 3, while the F-measure as a function of the number of clusters is shown in Fig. 1. The best F-measure is achieved when combining perplexity with pragmatic constraint filtering namely *QAPPL+constraints* and *WebPPL+constraints*. Results are better than those achieved by the best in-domain corpora, namely the *Q&A* corpus. Note the relatively poor performance of the *Human-System* corpus since it consists of short utterances lacking context necessary to the grammar induction algorithm. Also constraints help performance when combined with a perplexity criterion, but less so when using the *Top-1*. An important results is that the perplexity criterion is effective even if no in-domain corpus is available, i.e.,*WebPPL+constraints* case.

7. Conclusions

We showed that unsupervised web harvesting can result in a corpus that is as good for grammar induction as a manually collected in-domain corpus. The methods presented are language and domain agnostic and at the same time scalable; the end result can easily be tailored to the size specified by the application needs. Instrumental to the success of the proposed method is the combination of perplexity filtering with pragmatic constraints automatically estimated from the web corpus.

Acknowledgements This work has been been partially funded by the PortDial project ("Language Resources for Portable Multilingual Spoken Dialog Systems") supported by the EU Seventh Framework Programme (FP7), grant number 296170.

8. References

[1] H. Meng and K.-C. Siu, "Semi-automatic acquisition of semantic structures for understanding domain-specific natural language queries," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 1, pp. 172–181, 2002.

[2] A. Pargellis, E. Fosler-Lussier, C. H. Lee, A. Potamianos, and A. Tsai, "Auto-induced semantic classes," *Speech Communication*, vol. 43, no. 3, pp. 183–203, 2004.

[3] Z. Harris, "Distributional structure," *Word*, vol. 10, no. 23, pp. 146–162, 1954.

[4] A. Pargellis, E. Fosler-Lussier, A. Potamianos, and C.-H. Lee, "A comparison of four metrics for auto-inducing semantic classes," in *Proc. of Automatic Speech Recognition and Understanding Workshop*, 2001.

[5] A. Pangos, E. Iosif, A. Potamianos, and E. Fosler-Lussier, "Combining statistical similarity measures for automatic induction of semantic classes," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop*, 2005, pp. 278–283.

[6] E. Iosif, A. Tegos, A. Pangos, E. Fosler-Lussier, and A. Potamianos, "Unsupervised combination of metrics for semantic class induction," in *Proc. of the IEEE/ACL International Workshop on Spoken Language Technology (SLT)*, 2006.

[7] E. Iosif and A. Potamianos, "A soft-clustering algorithm for automatic induction of semantic classes," in *Proc. of Interspeech*, 2007.

[8] R. Pieraccini and J. J. Huerta, "Where do we go from here? research and commercial spoken dialog systems," in *Proc. of SIGDIAL Workshop on Discourse and Dialog*, 2005.

[9] R. Sarikaya, "Rapid bootstrapping of statistical spoken dialogue systems," *Speech Communication*, vol. 50, no. 7, p. 580593, 2008.

[10] Y.-Y. Wang and A. Acero, "Rapid development of spoken language understanding grammars," *Speech Communication*, vol. 48, no. 3-4, p. 390416, 2008.

[11] B. Cramer, "Limitations of current grammar induction algorithms," in *ACL*, 2007.

[12] J. Kuhn, "Experiments in parallel-text based grammar induction," in *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, 2004.

[13] A. Sethy, S. Narayanan, and B. Ramabhadran, "Data driven approach for language model adaptation using stepwise relative entropy minimization," in *Proc. of ICASSP*, 2002.

[14] T. Misu and T. Kawahara, "A bootstrapping approach for developing language model of new spoken dialogue systems by selecting web texts," in *Proc. of Interspeech*, 2006.

[15] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proc. of the 40th Annual Meeting on Association for Computational Linguistics*, 2002, pp. 311–318.

[16] J. Gao, J. Goodman, M. Li, and K.-F. Lee, "Toward a unified approach to statistical language modeling for chinese," *ACM Transactions on Asian Language Information Processing (TALIP)*, vol. 1, no. 1, pp. 3–33, mar 2002.

[17] A. Bisazza, I. Klasinas, M. Cettolo, and M. Federico, "FBK @ IWSLT 2010," in *Proceedings of the seventh International Workshop on Spoken Language Translation (IWSLT)*, M. Federico, I. Lane, M. Paul, and F. Yvon, Eds., 2010, pp. 53–58.

[18] T. Ng, M. Ostendorf, M. Hwang, M. Siu, I. Bulyko, and X. Lei, "Web data augmented language models for mandarin conversational speech recognition," in *IEEE International Conference on In Acoustics, Speech, and Signal Processing*, 2005, pp. 589–592.

[19] Lingua-Sentence-1.04, http://search.cpan.org/ achimru/Lingua-Sentence-1.04/.

[20] P. Clarkson and R. Rosenfeld, "Statistical language modeling using the cmu-cambridge toolkit," in *Proc. of Eurospeech*, 1977.

[21] CLUTO: A Clustering toolkit, http://glaros.dtc.umn.edu/gkhome/cluto/cluto/overview.