A BIN-BASED ONTOLOGICAL FRAMEWORK FOR LOW-RESOURCE N-GRAM SMOOTHING IN LANGUAGE MODELLING

Y. Benahmed[1,2], S.-A. Selouani[2] and D. O'Shaughnessy[1]

[1]INRS-EMT, 800 de la Gauchetière O, H5A 1K6, Montréal, Quebec, Canada
[2]LARIHS Lab. Université de Moncton, campus de Shippagan , New-Brunswick, Canada
yacine@umcs.ca, selouani@umcs.ca, dougo@emt.inrs.ca

ABSTRACT

In this paper, we introduce a novel method of smoothing language models (LM) based on the semantic information found in ontologies that is especially adapted for limited-resources language modeling. We exploit the latent knowledge of language that is deeply encoded within ontologies. As such, this work examines the potential of using the semantic and syntactic relations between words from the WordNet ontology to generate new plausible contexts for unseen events to simulate a larger corpus. These unseen events are then mixed-up with a baseline Witten-Bell (WB) LM in order to improve its performance both in terms of language model perplexity and automatic speech recognition word error rates. Results indicate a significant reduction in the perplexity of the language model (up to 9.85% relative) all the while reducing word error rate in a statistically significant manner compared to both the original WB LM and baseline Kneser-Ney smoothed language model on the Wall Street Journal-based Continuous Speech Recognition Phase II corpus.

Keywords: Language modeling, context modeling, ontologies, low-resource speech recognition

1. INTRODUCTION

Language modeling is an integral part of the speech recognition task where the purpose is to try to reliably predict what the speaker is most likely to say next. Today, *n*-gram language modeling remains the dominant technique in automatic speech recognition applications. Formally, we compute the probability of word $w = w_n$ given its history or context h : $P(w|h)$ where $h = w_{n-1}, w_{n-2}, ..., w_{n-N}$ is obtained from counts of very large textual corpus. However, this is not always convenient or practically achievable since new sentences are constantly being created [1]. For example it would be impossible to predict a novel sentence, since its count would be zero. In addition, it is easy to see that as the length of the sentence and/or vocabulary grows, the task of obtaining those counts becomes impractical. The goal of *n*-gram language models then is to simplify the prediction of the *n*-th word based on the $n-1$ previous words using maximum likelihood estimates (MLE):

$$P_{MLE}(w_n|h) = \frac{C(h, w_n)}{C(h)}. \tag{1}$$

Two well-known drawbacks of language modeling with *n*-grams are its sensitivity to the corpus and data sparseness, i.e. if we train our model on text from Shakespeare, it is evident that the model will not be adequate for the prediction of astrophysical text, namely the count of the word "magnetohydrodynamics" in a Shakespearian work $C(magnetohydrodynamics_{astrophysics}|h_{Shakespeare})$ is surely equal to 0. That is why multiple approaches exist to either obtain better language models (LM) or smooth existing ones in order to improve their statistical accuracy and generalizability. Notable techniques include discounting, interpolation and various backoff schemes such as Good-Turing Discounting, Witten-Bell discounting [2], Modified Kneser-Ney smoothing [3] or Hierarchical Pitman-Yor language models [4] used to smooth out low-order counts. Variable length n-grams and class-based n-grams [5] also enabled better performance of language models. Other work such as [6, 7, 8, 9] looks into exploiting the vast amount of text available on the Web and raw n-gram counts such as those provided by Google.

Although we have made significant leaps in the area there is still room for progress. This is particularly true in the case where we don't have access to training data that relates to the testing or real-world data or that such resources are limited and/or too expensive. It is for these reasons that this work mainly focuses on semantically expanding the information found in small corpus for smoothing language models.

In this work, we introduce a novel method of smoothing language models by exploiting the semantic information found in ontologies. It will be used in combination with Witten-Bell smoothing since the intuition behind this smoothing algorithm is to look at the diversity of contexts in which a particular history occurs: how likely is it to see a new n-gram given a particular history. This is especially suited for our n-gram count smoothing algorithm since one of its goals is to generate a new set of probable n-gram events that enrich the contextual diversity of the language model.

The contribution of this work is the demonstration that ontological relations can reliably be used for smoothing existing n-gram counts as well as producing novel n-gram contexts to enrich the language model and increase its performance both in terms of perplexity and in an automatic speech recognition task.

The outline of this paper is as follows. Section 2 introduces ontologies and the WordNet ontology used in this work. Section 3 describes our proposed ontological n-gram count smoothing algorithm. Section 4 proceeds with the description of the experiment set-up and the evaluation of our algorithm. Finally, in section 5, we conclude and discuss our results.

2. ONTOLOGIES

Ontologies are used to relate concepts in a hierarchical and relational fashion that ultimately describe the meaning of terms. Logical and philosophical entities can be grouped contextually with the use of named relations. Figure 1 shows a portion of the en-

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

try for the ontology of a cat as it relates to rodents and birds. In this example, solid lines represent inheritance, dashed lines such as the ones connecting cat with bird and rodents represent functional relationships and the dotted lines between vertebrate and spine represent inherited properties.

2.1. The WordNet ontology

We used the popular WordNet [10] as our base ontology from Princeton University. It is essentially a large lexical database of English. The main relation between defined words is synonymy, the state of being a synonym [10]. WordNet lemmas are classified by part of speech: adjectives (21,479), adverbs (4,481), nouns (117,798) and verbs (11,529) for a total of 155,287 senses; however since many words are found in multiple parts-of-speech, a total of 147,278 unique indexed words are found in the ontology.

3. ONTOLOGY-BASED *N*-GRAM SMOOTHING

In this section we introduce our novel technique to improve the performance of *n*-gram smoothing. One of the problems with current language modeling techniques, as previously stated, is that they do not necessarily generalize themselves to words of the same semantic class/sub-class. Counts for words such as cat, dog, mouse, etc., all being animals, could be proportionally smoothed by the *n*-gram entry for *lamb* from the example of section 2.2. The same could be said about relational markers between objects where similar relations could be smoothed with the help of well established ones. As such, we propose a method of smoothing the *n*-gram counts by leveraging the latent information stored in ontologies. Ontologies are used to relate concepts in a hierarchical and relational fashion. As was stated in section 2, entities can be grouped logically and/or philosophically in a contextual fashion through the use of named and weighted relations. Figure 1 shows a portion of the WordNet ontology where the ontological distance from `cat` = 1.

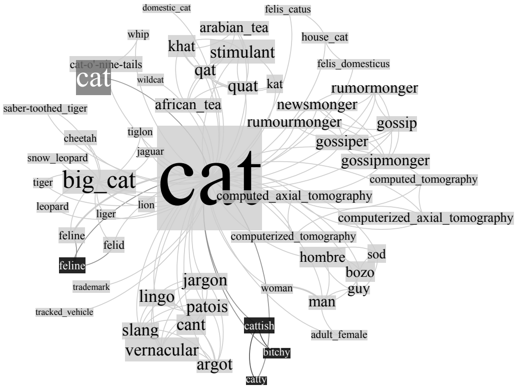

Fig. 1. Sample of the full ontology as related to a cat. The size of the nodes represent their degree of connectedness and their color their part of speech class: adjectives (black), adverbs (not shown, white), nouns (light gray) and verbs (gray). The colors of the edges represent the type of relation between nodes. Note that the length of the lines have no meaning other than to improve the clarity of the illustration.

Therefore, by studying the information contained within, we are able to smooth our *n*-gram language model by smoothing its probabilities with those of weighted *n*-grams of related terms, i.e., use the counts of related *n*-grams $C(w_r|h)$ for each word $w_r \in W_R$ related to w_n and its context h to smooth the counts $C(w_n|h)$, where W_R is a vector of R words related to w_n. Inversely, to smooth out zero counts, we can use the information from known $C(w_n|h)$ counts to generate semantically probable unseen $C(w_r|h)$ smoothed counts.

We believe that this is a reasonable assumption, especially for spoken dialogue where speakers can interchange words for related terms that come to mind more quickly. Furthermore, since this work seeks to help reduce the data sparseness of counts that are used for smoothing we propose the following technique based on interpolation to smooth low-resource language models by intelligently adding information not seen before.

3.1. Bin-based Ontological Smoothing

The bin-based ontological smoothing language model (BBOSLM) smoothing technique consists of creating bins B_d of smoothed counts taking into account all words found at a distance d, the shortest path (number of edges) in the ontology from words w_n and w_r, from the original *n*-gram counts and where $d \leq 1 \leq d_{max}$ and d_{max} is the maximal distance from w_n that is considered in the ontology. For example, in figure 1, the distance $d(\texttt{cat}, \texttt{tiger}) = 1$, $d(\texttt{feline}, \texttt{cattish}) = 2$, etc. More formally, for each $C(w_n|w_{n-N+1}^{n-1}) \in B_d$:

$$C(w_n|w_{n-N+1}^{n-1}) = \frac{1}{R} \sum_{r=1}^{R_d} \frac{C(w_r|w_{n-N+1}^{n-1})}{d} \tag{2}$$

where R is the total number of words related to w_n up to d_{max}. In practice, a vector of related words is pre-computed at the start of the program using a modified form of depth-limited iterative deepening depth-first search. As such, R is the number of linked terms for each word and $R_d \leq R$ is the number of related terms at distance d. The premise behind the use of the shortest path as a measure of conceptual or semantic distance is illustrated in [11], namely that "*...when is-a hierarchies are defined ... shortest path length can be used to measure conceptual distance between concepts.*" The ontologically generated counts are then used to generate language models for each bin LM_{B_d}. Finally, these language models are interpolated into the final smoothed BBOSLM:

$$\begin{aligned} P'(w_n|w_{n-N+1}^{n-1}) &= \lambda_1 P(w_n|w_{n-N+1}^{n-1}) \\ &+ \sum_{d=1}^{D} \lambda_{d+1} P_{B_d}(w_n|w_{n-N+1}^{n-1}) \end{aligned} \tag{3}$$

where $D = d_{max}$ is the maximum edge distance from the original in the ontology, $P_{B_d}(w_n|w_{n-N+1}^{n-1})$ are the probabilities obtained from the smoothed counts in each bin and where λ are the mixture weights and

$$\sum_{i=1}^{D+1} \lambda_i = 1$$

are obtained with the help of Powell optimization.

3.2. Ontological smoothing algorithm

The first step in the ontological smoothing process is to convert the ontology to an efficient graph format for processing. Given an ontology O (WordNet in this case) and part of speech classes POS, we convert the ontology to a directed graph G where each node

N is a lexeme (lemma + forms). Each node can be from multiple classes, e.g. `cat` is both a noun and a verb and each edge E represents a relation between two nodes.

The second step is to tag our corpus's terms by their part-of-speech. For this task, we used the Penn-State TreeTagger [12]. This is to preserve the meaning of the n-gram counts. For example, we potentially wouldn't want to smooth the n-gram entry of a verb with that of a noun. As per [11] we restrict the search space to synonymous terms for this work. Once the corpus is tagged, we proceed with the counting task using the SRILM toolkit [13].

Our algorithm operates in D passes: each pass creates a bin containing the smoothed counts for distance d using (2). Each pass also adds unseen $w_r|h$ events in order to reduce the effects of limited corpus. Since the number of new counts generated can be particularly high given a large distance metric, we parallelized our algorithm. This gives us a speed-up of up to 52% on four threads with $d_{max} = 5$ for the whole smoothing process.

3.3. Ontology search algorithm for R related terms

Since the search space of the ontology is considerable we needed an efficient algorithm to look for the related terms of each n-gram entry. The fact that we use an unweighted graph enabled us to avoid the inherent challenges of searching weighted directed graphs. The algorithm is a depth-limited form of unweighted iterative deepening depth-first search. Basically, as we progress through the graph each relation is marked gray; we then recursively explore each unmarked child node (we do not return to it since it has already been seen at a lower level), and thus the "shortest" path to it is already known. This greatly reduces the cost of exploring the ontology all the while guaranteeing optimal paths for each relation.

4. EXPERIMENTS AND RESULTS

4.1. Limited Resources Experimental Setup

The experiments reported in this paper were performed using the Wall Street Journal-based Continuous Speech Recognition Phase II corpus (WSJ1). We train our language models on the 76,136 sentences provided by the standard WSJ1 training sets for a total of 1,243,340 words. For the evaluation we restricted ourselves to the Hub 1 and Spoke 1 tests (10,347 words in 582 sentences) without filtering out sentences with out-of-vocabulary words (OOV rate of 5.41%).

The SRILM toolkit and HTK toolkit [14] were used for our experiments for language modeling and automatic speech recognition respectively. We obtain raw training counts using the SRILM toolkit. These raw counts are then used to generate ontologically smoothed count bins for each distance group, note that, as per eq. 2, the majority of the counts are fractional counts. Then, Witten-Bell LMs were created for each bins. Finally, a mixture-based language model is created by mixing up each bin-based LM with the original Witten-Bell LM. Optimal mixup weights were obtained through the use of Powell's optimization algorithm [15] using the SciPy library [16] with the objective function defined as the word error rate of the held-out development set consisting of 4,340 sentences for a total of 71,759 words. Unfortunately, the SRILM toolkit does not support fractional counts for Kneser-Ney smoothing and as such, we cannot properly interpolate the baseline KN model with our bins.

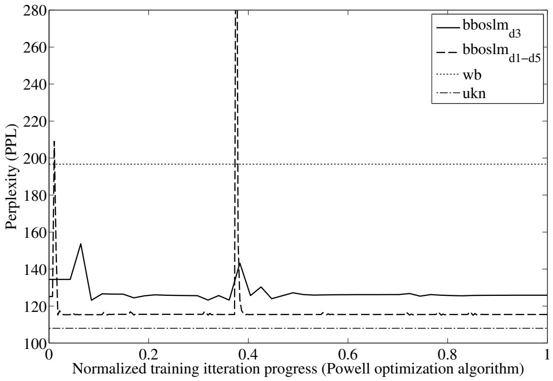

Fig. 2. Perplexity evolution during the Powell WER optimization process to determine the best BBOSLM bin mixup weights for 20K word trigram LMs trained on the WSJ1 training corpus and conditioned the WSJ1 development corpus. The spikes are due to the algorithm trying to see if it is in a local optimum.

Perplexity and word error rate (WER) are used to measure the performance of our language modeling technique. For the acoustic models, we use the well-known recipe from Vertanen [17]. Our model is trained by using all of the WSJ1 training data using the 40 phones set of the CMU dictionary. Cross-word tied-state triphones with 32 Gaussian mixtures per state and 64 Gaussian mixtures per silence state are eventually generated. The acoustic models are represented by Mel Frequency Cepstral Coefficients with energy, delta and acceleration ($MFCC\_E\_D\_A$) for a total of a 39-dimensional feature vector. The HDecode tool with parameters for the pruning beam width, word insertion penalty, and the language model scale factor of 220.0, -4.0, and 15.0 respectively were used for the automatic speech recognition test.

4.2. Perplexity evaluation

Perplexity evaluation was performed on the Hub 1 (read WSJ baseline) and Spoke 1 (language model adaptation) portion of the WSJ1 corpus which consist of 582 sentences and 10,347 words. The standard 20,000-word WSJ closed-vocabulary was used for the evaluation and backed-off trigram language models were generated using SRILM with the -float-counts option enabled.

The average path length of the ontology is 59.24 with a network diameter of 15 measured using the Gephi toolkit [18]. The network diameter is a measure of the maximum distance between any two pairs in the graph. Since it would be too computationally intensive to fully explore each node, we chose to study the effects of d_{max} from 1 to 5. From the ~ 346,000 original raw counts we end up with ~ 15,284,000 smoothed counts for B_5. Figure 2 shows the evolution of the language model perplexity throughout the Powell optimization algorithm for the Bin-based ontological smoothing against the baseline models trained using Witten-Bell (WB) smoothing and the original Kneser-Ney (UKN) smoothing. Note that we were not able to include results for Modified Kneser-Ney (MKN) smoothing as the SRILM toolkit kept computing negative discount parameters for the trigrams. Results for both single and multiple bins mixup are provided. Table 1 shows a summary of the best results obtained. As was expected, our technique is able to significantly reduce the perplexity of the baseline language models with the best model showing maximal perplexity reduc-

Table 1. Comparison of perplexity on WSJ1 Hub 1 and Spoke 1 evaluation corpus using 20K word trigrams using best WER performing single and multiple bins mix-up.

Language Model	**Perplexity**
Witten-Bell	142.61
Kneser-Ney	117.15
BBOSLM	-
$d = 1$	137.46
$d = 2$	137.5
$d = 3$	137.92
$d = 4$	137.54
$d = 5$	137.53
$1 \leq d \leq 2$	137.14
$1 \leq d \leq 3$	128.55
$1 \leq d \leq 4$	129.45
$1 \leq d \leq 5$	**122.49**

tion of 9.85% for the WB LM. We were able to get very close to the perplexity of Kneser-Ney discounting for Bins $1 \leq d \leq 5$. However this is to be expected as the training process did not explicitly look at minimizing perplexity but rather minimizing Word Error Rates. This is consistent with the literature as it is well known that a reduction in perplexity will not always translate in improvements in automatic speech recognition tasks. The spikes in perplexity are due to the Powell algorithm exploring the bounded space to see if it is not currently in a local optimum. This is encouraging since it demonstrates that our technique can compete with state-of-the-art language modeling techniques.

4.3. Automatic speech recognition experiments

This experiment investigates the performance of the BBOSLM on word error rate in an automatic speech recognition task. Much like the perplexity evaluation, we provide the WER results for single bin mixup and multiple bins mixup in relation to the LM perplexity optimization process. The optimization process was run three times with evenly distributed starting λ values, figure 3 shows the optimization process for the best performing final models and starting with $\lambda_0 = 0.9$. Final λ_1 from eq. 3 values for the two best performing LMs, $BBOSLM_{d3}$ for single bin mix-up and $BBOSLM_{d1-d5}$ for multiple bin mix-up, are 0.9962 and 0.8913 respectively. Globally, results show that the best performing LM are the ones mixing-up multiple smoothed counts bins. These results are confirmed by every statistical significance test available in the NIST Scoring Toolkit: a statistically significant reduction (mean $p = 0.001$) in word error rate of 6.8% relative for $BBOSLM_{d1-d5}$ and statistically significant reduction (mean $p = 0.01$) for $BBOSLM_{d4}$ with a considerable reduction in insertion and substitution rates compared to the baseline Witten-Bell discounted LM.

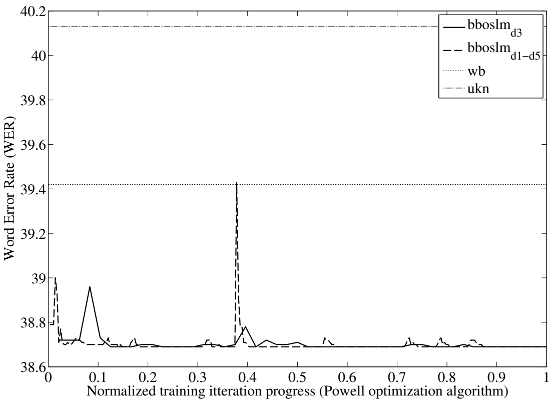

Fig. 3. WER (%) evolution of the BBOSLM during the WER optimization process on WSJ1 development corpus using 20K word trigrams. The "large" spike in the training process is due to the Powell algorithm jumping in λ values to see if it is in a local optimum.

Table 2. ASR performance comparison on WSJ1 Hub 1 and Spoke 1 evaluation corpus using 20K word trigrams using single and multiple bins mix-up.

LMS	**PPL**	**SUB**	**DEL**	**INS**	**WER**
WBLM	142.61	22.43	5.89	2.64	30.96
KNLM	117.15	21.87	7.71	1.41	30.98
BBOSLM	-	-	-	-	-
$d = 1$	137.46	21.20	6.65	1.72	29.56
$d = 2$	137.5	21.13	6.67	1.77	29.57
$d = 3$	137.92	20.99	6.67	1.78	29.44
$d = 4$	137.54	21.08	6.73	1.74	29.55
$d = 5$	137.53	21.05	6.79	1.73	29.57
$1 \leq d \leq 2$	137.14	21.18	6.65	1.73	29.56
$1 \leq d \leq 3$	128.55	20.69	6.76	1.58	29.03
$1 \leq d \leq 4$	129.44	20.72	6.74	1.6	29.06
$1 \leq d \leq 5$	122.49	20.23	7.03	1.59	28.85

5. CONCLUSION AND DISCUSSION

In this paper, we introduced an original method of preprocessing and smoothing n-gram counts based on the semantic information found in ontologies for use in limited resources automatic speech recognition. It exploits the relations found in ontologies to create bins of new n-gram events related to existing n-gram events. These bins are transformed into language models (LM) and then mixed-up with the original LM. Experiments demonstrate that our smoothing technique reduced the perplexity of a given language model by up to 9.85% when considering bins of related words with a maximal ontological distance of up to 5. In addition, we evaluated our language models on the Hub 1 and Spoke 1 tasks of the Wall Street Journal-based Continuous Speech Recognition Phase II corpus. We showed that it reduced word error rate (WER) in a statistically significant manner compared to Kneser-Ney discounting ($p = 0.001$). The fact that we were able to obtain statistically significant performance increase indicate that it shows promise as a tool for language model smoothing, especially when textual resources are sparse. Future work will look to incorporate different ontological distance metrics such as those proposed by Jing et al.[19] as well as exploring information theory-based schemes to weight the different relations between words described by the ontology.

6. REFERENCES

[1] D. Jurafsky and J. H. Martin, *Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.* Prentice Hall, 2008.

[2] I. H. Witten and T. C. Bell, “The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression,” *Information Theory, IEEE Transactions on*, vol. 37, no. 4, pp. 1085–1094, 1991.

[3] S. F. Chen and J. Goodman, “An empirical study of smoothing techniques for language modeling,” in *Proceedings of the 34th annual meeting on Association for Computational Linguistics.* Association for Computational Linguistics, 1996, pp. 310–318.

[4] S. Huang and S. Renals, “Hierarchical bayesian language models for conversational speech recognition,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 18, no. 8, pp. 1941–1954, 2010.

[5] W. Naptali, M. Tsuchiya, and S. Nakagawa, “Topic-dependent-class-based -gram language model,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 5, pp. 1513 –1525, july 2012.

[6] X. Zhu and R. Rosenfeld, “Improving trigram language modeling with the world wide web,” in *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP ’01). 2001 IEEE International Conference on*, vol. 1, 2001, pp. 533–536 vol.1.

[7] I. Bulyko, M. Ostendorf, and A. Stolcke, “Getting more mileage from web text sources for conversational speech language modeling using class-dependent mixtures,” pp. 7–9, 2003.

[8] A. Sethy, P. G. Georgiou, and S. Narayanan, “Building topic specific language models from webdata using competitive models,” pp. 1293–1296, 2005.

[9] D. Lin, K. Church, H. Ji, S. Sekine, D. Yarowsky, S. Bergsma, K. Patil, E. Pitler, R. Lathbury, V. Rao *et al.*, “New tools for web-scale n-grams,” in *Proceedings of LREC*, 2010.

[10] G. A. Miller, “Wordnet: A lexical database for english,” *Communications of the ACM*, vol. 38, pp. 39–41, 1995.

[11] J. H. Lee, M. H. Kim, and Y. J. Lee, “Information retrieval based on conceptual distance in is-a hierarchies,” *Journal of documentation*, vol. 49, no. 2, pp. 188–207, 1993.

[12] H. Schmid, “Probabilistic Part-of-Speech Tagging Using Decision Trees,” in *Proceedings of the International Conference on New Methods in Language Processing*, 1994, pp. 44–49. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.28.1139

[13] A. Stolcke, “SRILM - An Extensible Language Modeling Toolkit,” 2002, pp. 901–904.

[14] S. Young, “The HTK Hidden Markov Model Toolkit: Design and Philosophy,” *Entropic Cambridge Research Laboratory, Ltd*, vol. 2, pp. 2–44, 1994.

[15] M. J. Powell, “An efficient method for finding the minimum of a function of several variables without calculating derivatives,” *The computer journal*, vol. 7, no. 2, pp. 155–162, 1964.

[16] E. Jones, T. Oliphant, P. Peterson *et al.*, “SciPy: Open source scientific tools for Python,” 2001–. [Online]. Available: http://www.scipy.org/

[17] K. Vertanen, “Baseline WSJ acoustic models for HTK and Sphinx: Training recipes and recognition experiments,” http://www.keithv.com/pub/baselinewsj, Cavendish Laboratory, University of Cambridge, Tech. Rep., 2006.

[18] M. Bastian, S. Heymann, and M. Jacomy, “Gephi: An open source software for exploring and manipulating networks,” 2009. [Online]. Available: http://www.aaai.org/ocs/index.php/ICWSM/09/paper/view/154

[19] L. Jing, L. Zhou, M. K. Ng, and J. Z. Huang, “Ontology-based distance measure for text clustering,” in *Proceedings of the Text Mining Workshop, SIAM International Conference on Data Mining*, 2006.