Evaluation of Graph Metrics for Optimizing Bin-Based Ontologically Smoothed Language Models

Yacine Benahmed*,†, Sid-Ahmed Selouani†, Douglas O'Shaughnessy*
*INRS-EMT, Montréal, Quebec, Canada
†LARIHS Lab. Université de Moncton campus de Shippagan, Shippagan, New-Brunswick, Canada
Email: yacine.benahmed@umoncton.ca, sid-ahmed.selouani@umoncton.ca, dougo@emt.inrs.ca

***Abstract*—This paper investigates the use of graph metrics to further enhance the performance of a language model smoothing algorithm. Bin-Based Ontological Smoothing has been successfully used to improve language model performance in automatic speech recognition tasks. It uses ontologies to estimate novel utterances for a given language model. Since ontologies can be represented as graphs, we investigate the use of graph metrics as an additional smoothing factor in order to capture additional semantic or relational information found in ontologies. More specifically, we investigate the effect of HITS, PageRank, Modularity, and weighted degree, on performance. The entire power set of bins is evaluated. Our results show that the interpolation of the original bins at distances 1, 3 and 5 resulted in an improvement in WER of 0.71% relative over the interpolation of bins 1 to 5. Furthermore, modularity, PageRank and HITS show promise for further study.**

I. Introduction

Language modeling is a crucial part of automatic speech recognition (ASR) systems as well as of many other tasks such as part-of-speech tagging, natural language processing, document classification, information retrieval, etc. More specifically for ASR however, language modeling enables recovering the probability of a sentence or sequence of words $S = w_1, \dots, w_i$ given an acoustic signal A, that is

$$P(S|A) = \frac{P(A|S)P(S)}{P(A)},$$

where $P(S)$ represents the language model (LM), the probability of the sentence S. The goal of n-gram language models is to simplify the prediction of the i-th word based on the $n-1$ previous words, essentially a Markov chain of order $(n-1)$, using maximum likelihood estimates (MLE):

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

A. Motivation

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. 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 significance. Notable techniques include discounting, interpolation and various backoff schemes such as Good-Turing Discounting [1], [2], Witten-Bell smoothing [3], Modified Kneser-Ney smoothing [4] or Hierarchical Pitman-Yor language models [5] used to smooth out low-order counts. Another drawback of n-gram language models is that they only can capture natural language semantics to a certain extent, mainly by capturing direct co-occurrences which do not completely reflect the co-occurrence networks from natural language. This is well documented in work by Biemen *et al.* [6]. Furthermore, work by Yan *et al.* [7] provide interesting insights on the use of ontologies in language modeling but only concentrates on document retrieval task. This is why bin-based ontologically smoothed language models (BBOSLM) [8] were proposed to exploit the semantic information found within the WordNet [9] ontology.

Graph theory, or network analysis, has gathered a lot of research interest over the last few decades. It can be seen as a powerful tool to quantify and qualify the importance and influence of nodes within a graph or detect hidden communities of interest. It is widely used in natural language processing for modeling relations between words. It is this ability that we are interested in examining in this work. We propose to study the effect of different network metrics used as additional weighting factors to the distance based similarity measure used in the original BBOSLM.

B. Contribution

The contribution of this work is the evaluation of the use of graph measures in ontology distance optimisation between words in BBOSLM. Furthermore, whereas previous work restricted itself to evaluating a linear incrementation of bins combination, we extend this work by evaluating the entire powerset of bins interpolated with the original Witten-Bell smoothed language model. That is, we look at the entire power set (155 distinct combinations) of bins (described in Section II below) interpolated with the original language model. We also extend the evaluation to the Spokes 2 and 9 of the Wall Street Journal Continuous Speech Recognition Phase II corpus for a total of 1285 sentences up from the 582 sentences used in the original work.

978-0-9928-6265-7/16/$31.00 ©2016 IEEE

The outline of this paper is as follows. In Section 2, network analysis and the graph metrics of interest are briefly presented. Section 3 describes the bin-based ontological n-gram count smoothing algorithm with the additional α weighing factor. Section 4 proceeds with the description of the experiment set-up and the evaluation of the use of different network metrics. Finally, in section 5, we conclude and discuss our results.

II. Bin-based Ontological Smoothing

Bin-based Ontological Smoothing [8] was proposed in order to smooth counts based on the ontological distance d between words found in an ontology, namely the WordNet ontlology. As stated in the introduction, the intuition behind this proposed technique was to exploit the relations found between words in ontologies to try to capture the fact that a given message could be expressed in different ways and that similar messages could be expressed in a similar way. For example, “the dog barked” and “the bird sang” are both very similar and different at the same time. For this, we consider the relations between words in the WordNet as an “is-a” hierarchy. This makes it possible to assume that the shortest path between words can be used as a simple measure of conceptual/semantical distance[10].

In Bin-Based Ontological Smoothing, we begin by generating smoothed bins $B = \{B_1, B_2, ..., B_{d_{max}}\}$ of smoothed n-gram counts that take into account all of the related words $\{(w_i, w_r) : \min(d(w_i, w_r)) = d\}$, completing the n-gram context w_{i-n+1}^{i-1}. Then, let

$$B_d = \{(w_{i-n+1}^{i}, w_r | w_{i-n+1}^{i-1}) : \min\{\mathrm{d}(w_i, w_r) = d\}\}$$

be the bin B_d of smoothed n-grams where w_{i-n+1}^{i} is the original n-gram, $w^r | w_{i-n+1}^{i-1}$ is the set of related n-gram with ontological distance d between the original word w_i and related word w_r. We than generate smoothed counts following equation 2:

$$C_{B_d}(w_i | w_{i-n+1}^{i-1}) = \frac{1}{dR} \sum_{w_r} C(w_r | w_{i-n+1}^{i-1}) \quad (2)$$

where R is the total number of words related to w_i up to the maximal distance d_{max} such that $R = |\{(w_i, w_r) : \mathrm{d}(w_i, w_r) \leq d_{max}\}|$ and d is the current distance. In practice we pre-compute a vector of related words for each w_i up to the maximum distance of interest d_{max} with a modified depth-limited iterative deepening depth first search. This greatly reduces the computational complexity of generating the smoothed counts. We then use those new smoothed counts to derive statistical language models using available complimentary smoothing techniques. However, previous work [8] found that Witten-Bell smoothing produced better overall results than Kneser-Ney-based smoothing. Finally, these language models are linearly interpolated into the final smoothed Bin-based Ontological Smoothed LM:

$$P'(w_i | w_{i-n+1}^{i-1}) = \lambda_0 P(w_i | w_{i-n+1}^{i-1}) + \sum_{d=1}^{d_{max}} \lambda_d P_{B_d}(w_i | w_{i-n+1}^{i-1}) \quad (3)$$

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

$$\sum_{i=0}^{d_{max}} \lambda_i = 1$$

can be obtained with the help of any linear optimization algorithm.

A. 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 an undirected 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 the corpus's terms by their part-of-speech. For this task, we used the Penn-State Tree-Tagger [11]. This is to preserve the meaning of the n-gram counts. For example, we want to avoid smoothing the n-gram entry of a verb with that of a noun. As per [10] 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 .

The smoothing algorithm then operates in d_{max} passes: each pass creates a bin B_d containing the smoothed counts for distance d using (2). As stated in section II each pass also adds unseen $w_r | h$ events in order to smooth zero count n-grams.

B. Network Analysis

Graph theory or network analysis has gathered a lot of research interest over the last few decades. It can be seen as a powerful tool to quantify and qualify the importance and influence of nodes within a graph or detect hidden communities of interest. Its use in many areas of research, be it biology, sociology, economy etc. speaks volumes of its versatility in modeling relations. It is this ability that we are interested in studying in this work. We propose to study the effect of different network metrics used as additional weighting factors to the distance based similarity measure used in the previous work. More specifically, we investigate if using HITS Centrality, PageRank Centrality, Modularity, and a baseline weighted degree can improve the performance of the smoothing method.

1) Weighted Degree: Degree and its ponderated form, Weighted degree, are the simplest measures of centrality in graphs. It essentially counts the number of edges incident to a node and is defined as:

$$WD_i = \sum_j A_{ij} \quad (4)$$

where WD_i is the weighted degree of edge i, the sum of the weight of all edges attached to it, and Aij is the edge weight between nodes i and j. This equation can easily be rewritten with respect to a word network as follows:

$$WD(w_i) = \sum_{w_j} A(w_i, w_j). \tag{5}$$

That is, $WD(w_i)$ is the weighed degree of word w_i and $A(w_i, w_j)$ is the edge weight between words w_i, w_j.

2) HITS: The HITS (Hyperlink-Induced Topic Search) algorithm [12] rates Web pages by analysing their links. It gives a list of hub and authority centralities for each node in a graph. The relation between hubs and authorities is maintained by the mutually recursive algorithm

$$auth(p) = \sum_{i=1}^{n} hub(i), \quad hub(p) = \sum_{i=1}^{n} auth(i) \tag{6}$$

,which can be rewritten as:

$$auth(w_i) = \sum_{w_j} hub(w_j), \quad hub(w_i) = \sum_{w_j} auth(w_j), \tag{7}$$

where $auth(w_i)$ and $hub(w_i)$ are the authority and hub scores of word w_i based respectively on the hub and authority score of connected words w_j (inbound for authority and outbound for hub). That is, good hubs point to authoritative words and authoritative words are pointed by good hubs. Each node starts with a value of 1 for both authority and hub, we update $auth$ values followed by hub values. They are then are normalized respectively by the square root of the sum of their squares and the process repeats until the variation falls bellow a set threshold.

3) PageRank: The PageRank [13] algorithm is an iterative algorithm that gives the probability of a random surfer clicking its way randomly to any given web page. We assume word w_i has words $w_j ... w_{j+n}$ which point to it. The PageRank of a word w_i is then given as follows:

$$PR(w_i) = \frac{1-\delta}{N} + \delta \left(\sum_{w_j \in M(w_i)} \frac{PR(w_j)}{L(w_j)} \right), \tag{8}$$

where $PR(w_i)$ is the PageRank of word w_i and $L(w_j)$ is the number of words linked by w_j. The parameter δ is a damping factor which can be set between 0 and 1 and is usually set to $\delta = 0.85$, $w_j \in M(w_i)$ is defined as the set of words linked by word w_i and N is the total number of words in the vocabulary.

One advantage of the PageRank algorithm is that it can also be applied to undirected graphs. The assumption of this work is that a low PageRank suggests a low probability of substitution of word w_i for word w_j and is then used as a weighing factor with $0 \leq PR \leq 1$.

4) Modularity: Modularity is a technique used for the detection and characterization of community structure in any given network. The quality $-1 \leq Q \leq 1$ of such communities c is defined by their Modularity defined in [14] such that:

$$Q = \frac{1}{2m} \sum_{w_i, w_j} \left[A_{ij} - \frac{D(w_i)D(w_j)}{2m} \right] \sigma(c(w_i), c(w_j)), \tag{9}$$

where Aij is the edge weight between words w_i and w_j, $D(w)$, the (unweighted) degree of word w, the σ-function $\sigma(u, v)$ is 1 if $u = v$ and 0 otherwise (in other words, 1 if they are in the same community, 0 otherwise). For this work, modularity is used to penalize words that are from different communities c, that is, when $c(w_i) \neq c(w_r)$. We use it to try to uncover unknown topics or lemma classes within the ontology and penalize counts from words in different communities.

C. The WordNet ontology

The WordNet [9] ontology is used in this work. It is a essentially a large lexical database of English. The main relation between defined words is synonymy, the state of being a synonym [9]. It was chosen due to it's well documented nature and widespread use in the literature.

D. Weighting adjustments

The basis for this work is the hypothesis that less important words (essentially less connected words) should contribute less to smoothing counts of n-grams and more important words should contribute more. For this work, we consider the ontology as an undirected graph and we use the previously mentioned measures from network theory to extract meaningful information. It is then used to compute adjustment weights from an α-function for each ontological relation between words w_i and w_r. As such, this work proposes the following modification to the original equation (2):

$$C_{B_d}(w_i | w_{i-n+1}^{i-1}) = \frac{1}{dR} \sum_{w_r} \alpha(w_r) C(w_r | w_{i-n+1}^{i-1}), \tag{10}$$

where $\alpha(w_r)$ is the adjustment weight for related word w_r for the given network metric function. It represents the importance of w_r within the ontology. Furthermore, for the special case of Modularity, we define the α-function as such:

$$\alpha_{Mod}(w_i, w_r) = \begin{cases} 1, & \text{if } c(w_i) = c(w_r) \\ \frac{1}{e}, & \text{otherwise.} \end{cases}$$

All graph metrics were computed using Gephi [15] and as such, the standard algorithms for HITS ($\epsilon = 1.0E-4$) and PageRank (damping factor $\delta = 0.85$) were used. A notable exception being the Modularity computation, which uses the fast algorithm for unfolding communities proposed by Blondel *et al.* [16]. It is an efficient algorithm that results in high modularity partitioning. To speed up computation time, all properties are pre-computed and stored in a hash-key based binary storage matrix.

III. Experiments and results

A. Experimental Setup

The experiments reported in this paper were performed using the Wall Street Journal-based Continuous Speech Recognition Phase II corpus (WSJ1). The training of our models uses the 76k sentences provided by the basic WSJ1

TABLE I
COMPARISON OF PERPLEXITY ON WSJ1 HUB 1, SPOKE 1, SPOKE 2 AND SPOKE 9 EVALUATION CORPORA USING 20K WORD TRIGRAMS USING SINGLE AND MULTIPLE BINS MIX-UP FOR EACH NETWORK METRIC

LMS	BBOSLM	BBOSLM Auth	BBOSLM Mod	BBOSLM PR	BBOSLM WD	WB	KN
original	-	-	-	-	-	149.11	**134.65**
$d = 1$	147.01	147.01	148.44	147.01	160.24	-	-
$d = 2$	145.11	145.11	147.65	145.11	159.47	-	-
$d = 3$	144.71	144.71	142.85	144.71	157.12	-	-
$d = 4$	142.66	142.66	141.86	142.66	156.08	-	-
$d = 5$	142.87	142.83	141.63	142.83	158.40	-	-
$1 \leq d \leq 2$	139.26	139.18	140.39	139.18	158.37	-	-
$1 \leq d \leq 3$	138.55	138.55	137.56	138.55	158.57	-	-
$1 \leq d \leq 4$	137.41	137.50	**136.27**	137.50	156.47	-	-
$1 \leq d \leq 5$	136.31	136.33	137.24	136.59	155.76	-	-

training sets for a total of 1.2M words. For the evaluation we used the Hub 1 (read WSJ baseline), Spoke 1 (language model adaptation), Spoke 2 (domain-independance) and Spoke 9 (spontaneous WSJ dictation) test data (23k words in 1285 sentences) without filtering out sentences with out-of-vocabulary words. The standard 20K words WSJ closed-vocabulary (OOV rate of 5.64%) was used for the evaluation and backed-off trigram language models were generated using SRILM with the `-float-counts` option enabled for the bins.

The SRILM toolkit [17] and HTK toolkit [18] were used for language modeling and automatic speech recognition respectively. We first generated the raw training counts using the SRILM toolkit. These raw counts are then used to generate ontologically smoothed floating point raw count bins for each distance group. Then, Witten-Bell LMs were created for each bin. Finally, a mixture-based language model was created by mixing up each bin-based LM with the original Witten-Bell LM. Mixup weights were obtained through the use of Powell's optimization algorithm [19] using the SciPy library [20] with the objective function defined as the perplexity (PPL) of the held-out development set, consisting of 4.3k sentences. We chose the Powell optimization algorithm instead of the EM algorithm provided by `compute-best-mix` script from the SRILM toolkit because of empirical observations where we consistently obtained better results by using the Powell algorithm. Secondly, the obtained mixture weights do not achieve the same level of performance (in terms of perplexity and ASR (without bayes)) that we get from the use of the Powell algorithm. Unfortunately, the SRILM toolkit does not support float counts for Kneser-Ney modeling and as such attempts to mix KN-based bins with the original KN LM resulted in severe performance degradation.

Perplexity (PPL) and Word Error Rate (WER) are used to compare the performance of the different language models. For the acoustic models, we used the recipe from Vertanen [21] with additional discriminative training (MPE criterion). Our model is trained by using all of the WSJ1 training data using the 40 phones set of the CMU dictionary. Full details of the configuration used can be found in [8]. 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 ASR test.

The average path length of the ontology is 59.24 with a network diameter of 15 measured using the Gephi toolkit. 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 study bins with words that are up to $d_{max} = 5$. From the ~ 346k original raw counts we end up with ~ 15.3M smoothed counts for $d_{max} = 5$.

B. Perplexity evaluation

Table I shows a summary of the best results obtained (due to the number of different combinations (155), only the best results are presented). These results show that our technique is able to significantly reduce the perplexity of the baseline Witten-Bell smoothed LM (WBLM) with the best model showing a maxium perplexity reduction of 9% relative for the interpolation of bins $B1$ to $B5$ and with only a difference of 1.2% relative to Kneser-Ney smoothing. Results from the LMs using graph metrics closely match the original results. The Authority and PageRank centrality slightly reduce perplexity compared to the original for single bin interpolation at $d = 4$ and produce equivalent results to each other. We believe that this is due to the similarity (mean difference of 1.79E-05) between the measures obtained from both algorithms. Modularity shows particular interest since it performs better than the original counts from $B3$ to interpolation of bins $B1, B2, B3, B4$. As for LMs with bins adjusted with the weighted degree measure, their performance is worse across the board, with a perplexity score higher than the original WBLM.

C. Automatic speech recognition experiments

This experiment evaluates the effect of the graph measures as well as the exploration of the entire powerset on BBOSLMs in the context of automatic speech recognition.

TABLE II
ASR PERFORMANCE COMPARISON OF THE BEST PERFORMING LMS ON WSJ1 HUB 1, SPOKE 1, SPOKE 2 AND SPOKE 9 EVALUATION CORPORA USING 20K WORD TRIGRAMS USING SINGLE AND MULTIPLE BINS MIX-UP.

LMS	**PPL**	**SUB**	**DEL**	**INS**	**WER**
WBLM	149.11	17.96	**4.32**	3.14	25.42
KNLM	**134.65**	17.88	5.63	2.29	25.79
BBOSLM					
$bins_{d1-d5}$	136.31	16.61	5.27	**2.22**	24.10
$bins_{d1,d3,d5}$	137.36	**16.53**	5.17	2.23	**23.93**
$auth_{d1,d3,d5}$	138.11	16.62	5.17	**2.22**	24.01
$mod_{d1,d5}$	137.65	16.58	5.14	2.26	23.98
$pr_{d1,d3,d5}$	138.11	16.62	5.17	**2.22**	24.01
$wdeg_{d5}$	158.40	17.00	4.88	2.43	24.31

We provide only the best results for each graph metric as well as the unmodified BBOSLM with $\alpha(m) = 1$. Results from the LM combining unmodified bins 1 to 5 are also included for comparison purposes. The BBOSLM using only the three bins B1, B3, B5 gave the best results with a WER of 23.93% compared to a WER of 24.10% for the original BBOSLM using all five bins. This strongly suggests that not all bins contribute positively to the final language model. It is also interesting to see that better performance can be achieved by using fewer data. By looking at the other best performers, we see that the trend is maintained in the sense that bins B1, B3 and B5 are used in the majority of best performers. This behaviour deserves further investigation in future work. Even though the α weighted BBOSLMs did not outperform the best LM, they still perform better than both Witten-Bell and Kneser-Ney smoothed language models. Like with the perplexity experiments, weighted degree bins perform worse than the other graph metrics, yet this results in smaller word error rate than both KN and WB smoothing using only bin 5. We suspect that it might be symptomatic of a certain amount of robustness built into the BBOSLM smoothing algorithm.

IV. Conclusion and discussion

In this paper, we evaluated the impact of using different graph metrics to weigh the relations between words in the WordNet ontology for use with the Bin-Based Ontological Smoothing (BBOS) proposed in previous work [8]. We observed from our experiments that the hypothesis that less important words (network node) should contribute less to the overall smoothed counts of bins is inconclusive and should be the subject of further study. Although performance using network metrics slightly degraded compared to the best performing unmodified BBOSLM we were still able to obtain better performance than Witten-Bell and Kneser-Ney smoothing. Furthermore, we were able to obtain better performance by reducing the number of bins used during interpolation from five to three. This seems to indicate that only certain bins contribute positively to the smoothing of counts and again gives us other interesting avenues for future research.

References

[1] I. J. Good, "The population frequencies of species and the estimation of population parameters," *Biometrika*, vol. 40, no. 3-4, pp. 237–264, 1953. [Online]. Available: http://biomet.oxfordjournals.org/content/40/3-4/237.abstract

[2] W. A. Gale and G. Sampson, "Good-turing frequency estimation without tears," *Journal of Quantitative Linguistics*, vol. 2, no. 3, pp. 217–237, 1995.

[3] 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.

[4] 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.

[5] 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.

[6] C. Biemann, S. Roos, and K. Weihe, "Quantifying semantics using complex network analysis," in *In COLING'12.* Citeseer, 2012.

[7] R. Yan, H. Jiang, M. Lapata, S.-D. Lin, X. Lv, and X. Li, "Semantic v.s. positions: Utilizing balanced proximity in language model smoothing for information retrieval," in *Proceedings of the Sixth International Joint Conference on Natural Language Processing.* Nagoya, Japan: Asian Federation of Natural Language Processing, October 2013, pp. 507–515. [Online]. Available: http://www.aclweb.org/anthology/I13-1058

[8] Y. Benahmed, S.-A. Selouani, and D. O'Shaughnessy, "A bin-based ontological framework for low-resourcen-gram smoothing in language modelling," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, May 2014, pp. 4918–4922.

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

[10] 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.

[11] 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

[12] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM (JACM)*, vol. 46, no. 5, pp. 604–632, 1999.

[13] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine, 1998," in *Proceedings of the Seventh World Wide Web Conference*, 2007.

[14] M. E. Newman, "Analysis of weighted networks," *Physical Review E*, vol. 70, no. 5, p. 056131, 2004.

[15] 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

[16] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p. P10008, 2008.

[17] A. Stolcke, "SRILM - An Extensible Language Modeling Toolkit," 2002, pp. 901–904.

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

[19] 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.

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

[21] 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.