Expanded Lattice Embeddings for Spoken Document Retrieval on Informal Meetings

Esaú Villatoro-Tello*
Idiap Research Institute
Switzerland
esau.villatoro@idiap.ch

Srikanth Madikeri
Idiap Research Institute
Switzerland
srikanth.madikeri@idiap.ch

Petr Motlicek
Idiap Research Institute
Switzerland
petr.motlicek@idiap.ch

Aravind Ganapathiraju
Uniphore Software Systems Inc.
Palo Alto, CA, USA
aravindganapathiraju@uniphore.com

Alexei V. Ivanov
Uniphore Software Systems Inc.
Palo Alto, CA, USA
alexei_v_ivanov@ieee.org

ABSTRACT

In this paper, we evaluate different alternatives to process richer forms of Automatic Speech Recognition (ASR) output based on lattice expansion algorithms for Spoken Document Retrieval (SDR). Typically, SDR systems employ ASR transcripts to index and retrieve relevant documents. However, ASR errors negatively affect the retrieval performance. Multiple alternative hypotheses can also be used to augment the input to document retrieval to compensate for the erroneous one-best hypothesis. In Weighted Finite State Transducer-based ASR systems, using the n-best output (i.e. the top "n" scoring hypotheses) for the retrieval task is common, since they can easily be fed to a traditional Information Retrieval (IR) pipeline. However, the n-best hypotheses are terribly redundant, and do not sufficiently encapsulate the richness of the ASR output, which is represented as an acyclic directed graph called the lattice. In particular, we utilize the lattice's constrained minimum path cover to generate a minimum set of hypotheses that serve as input to the reranking phase of IR. The novelty of our proposed approach is the incorporation of the lattice as an input for neural reranking by considering a set of hypotheses that represents every arc in the lattice. The obtained hypotheses are encoded through sentence embeddings using BERT-based models, namely SBERT and RoBERTa, and the final ranking of the retrieved segments is obtained with a max-pooling operation over the computed scores among the input query and the hypotheses set. We present our evaluation on the publicly available AMI meeting corpus. Our results indicate that the proposed use of hypotheses from the expanded lattice improves the SDR performance significantly over the *n*-best ASR output.

CCS CONCEPTS

• **Information systems** → **Novelty in information retrieval**; **Language models**; • **Computing methodologies** → **Speech recognition**.

KEYWORDS

Speech retrieval; Informal spoken content search; Neural reranker; Neural language models; Lattice rescoring; Lattice expansion; Lattice embeddings

ACM Reference Format:
Esaú Villatoro-Tello, Srikanth Madikeri, Petr Motlicek, Aravind Ganapathiraju, and Alexei V. Ivanov. 2022. Expanded Lattice Embeddings for Spoken Document Retrieval on Informal Meetings. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3477495.3531921

1 INTRODUCTION

Increasing amounts of multimedia content, particularly spoken material, are being captured and archived from a wide variety of sources. However, the lack of robust retrieval systems to deal with the challenges present in spoken content limits the full potential of this material in any real-world application. Spoken Document Retrieval (SDR) systems usually employ a cascaded approach: the spoken document is processed by an Automatic Speech Recognition (ASR) followed by an Information Retrieval (IR) system that indexes the output of the ASR and processes all given queries [3, 35]. ASR performance is affected by speaking styles (e.g. conversational speech, broadcast data, meetings, etc.). The errors in ASR output in turn negatively affect the performance of IR systems that are usually trained with only clean textual data. Thus, the input to IR may be enriched by either using *n*-best hypotheses of the ASR system, consensus network, or even the pruned version of the decoded lattice for indexing. Alternatively, end-to-end approaches also exist that take both the speech document and query as input, and output relevance scores [6]. Such methods, however, require large amounts of training data in order to develop robust systems.

In this paper, we present an extensive analysis on different alternatives to process richer forms of ASR output for SDR on the AMI corpus. Our work has three salient features: (1) a neural reranking approach based on expanded lattice embeddings space; we evaluate and report IR results using two different alternatives to use

*Also with Dept. of Information Technologies, at Universidad Autónoma Metropolitana, Unidad Cuajimalpa, Mexico City, Mexico. (evillatoro@correo.cua.uam.mx).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SIGIR '22, July 11–15, 2022, Madrid, Spain
© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3531921

information in the lattice, (2) new baseline on the AMI corpus; we present our results on the AMI corpus prepared for SDR by [5] and establish a baseline for this dataset with ASR model trained on out-of-domain data, (3) we also do not use any part of the target dataset to train the IR reranker, allowing the possibility of our method to be domain-agnostic.

The remainder of the paper is organized as follows: §2 describes the fundamentals of ASR lattice-based outputs, §3 describes the proposed SDR method, §4 explains the employed dataset, baseline methods, how the sentence embeddings were obtained, and experiments definition. We discuss both the ASR performance and the SDR results in §5. Finally, §6 depicts our main conclusions and future work directions.

2 ASR LATTICE-BASED OUTPUTS

In Weighted Finite State Transducer (WFST)-based ASR systems, decoding converts the input speech to an intermediate representation known as the lattice [25]. The lattice is a directed graph with each edge, also known as an arc, containing information about acoustic model (AM) and language model (LM) scores. The path with the best score, also referred to as the one-best output, is considered as the ASR output. Since the lattice is generated with a statistical n-gram LM, it is generally better to apply LM rescoring (i.e. replace the language model scores in the lattice) with a significantly better LM (typically neural LMs) [33].

ASR outputs have been applied to many downstream tasks in NLP: Natural language understanding (e.g. intent detection, domain detection, etc.), information retrieval, keyword spotting, etc. The top-scoring ASR hypothesis (also referred to as the one-best) may contain errors and omissions whereas the downstream systems are often trained with ground truth data that are error-free. Such systems are not trained to be robust to error-prone inputs. For instance, machine translation (MT) applied to ASR's one-best outputs reduce BLEU scores significantly [29].

It is often necessary to enrich the input to these downstream tasks. The enrichment focuses on using alternate hypotheses available after decoding the speech document, which is well captured in the lattice. A simple approach is to use n-best outputs, i.e. top n-scoring paths in the lattice. However, the n-best hypotheses suffer from redundancy where consecutive outputs have a significantly large overlap. To ignore the redundancy it is possible to compress the information in a confusion network (also referred to as consensus network) [18]. A confusion network is obtained from a word lattice and provides multiple word hypotheses at different time instances with confidence value for each word. In [35], the confusion networks are indexed for IR. A similar approach is applied for Cross-lingual IR in [3, 35]. As an extension to indexing confusion networks, ASR lattices are also indexed with applications to keyword spotting [30].

In all-neural ASR systems it is now possible to plug neural IR and rerankers by jointly optimizing the two tasks [6]. However, it is not straightforward to use such models in hybrid ASR systems. While it is possible to simplify the process by using only the one-best output from the ASR, we miss out on the richness of the n-best and the lattice.

Contrary to previous work, in this paper, we propose to apply neural reranking for SDR on ASR lattices by using well-known methods for neural LM rescoring in hybrid ASR systems. Rescoring lattices with neural LMs are employed in two ways: (1) the n-best output is generated and the LM score is evaluated for each hypothesis with the neural model, and the hypothesis with the best combined–AM and LM–score is now chosen as the ASR output; (2) RNN LMs are applied on lattice paths by limiting the context [11, 12]. The LM scores generated with n-gram models are replaced by those provided by the RNN LMs. The best path is now re-computed to obtain the new ASR output.

We utilize the algorithm proposed in [10] that generates a minimal set of hypotheses that can be scored with neural LMs. The lattice is pruned and expanded, followed by the application of the constrained minimum path cover algorithm on the lattice. The constraints ensure every path is the best path for at least one arc. The minimum path cover produces a lattice containing the fewest possible paths. We will refer to the representations of each path from a neural LM as the expanded lattice embedding. The paths generated by this lattice expansion algorithm act as a proxy to score a target query against the lattice. Scoring all outputs (i.e. processing through the IR system) of the constrained minimum path algorithm ensures that all arcs in the lattice are entirely covered, even though we are not evaluating every path in the original lattice. We compare the expanded lattice embeddings against the embeddings from the n-best outputs with $n = 100$. Whereas in [10] the authors implemented LM rescoring for models trained with Pytorch on lattices output by Kaldi, we target the task of SDR for WFST-based ASRs. In addition to using the embeddings, we also integrate the ASR confidences in the reranking process as explained in the next section. We consider two types of scores to compute these confidences: (1) the Viterbi-Forward-Backward scores generated during lattice expansion, and (2) the final AM and LM scores in the lattice for each path considered by the lattice expansion algorithm. We note that, confidence scores generated with the second method will be equivalent to those generated by the n-best algorithm when both algorithms generate the same number of hypotheses. If the lattice expansion algorithm generated m paths for a given lattice, and if we consider the top-m scoring paths in the lattice, we did not notice any difference in the set of hypotheses generated. The order of the hypotheses was, however, different.

3 RERANKING WITH EXPANDED LATTICE EMBEDDINGS

Figure 1 summarizes our proposed multi-stage retrieval methodology. Given a query $q_i \in Q$ and a spoken document collection $\mathcal{D}$, the goal of the first stage retrieval is to find a set of documents relevant to the query $D^+ = \{d_1, ..., d_j, ..., d_n\}$ from $\mathcal{D}$ such that ($|D^+| \ll |\mathcal{D}|$), which then serve as input to our neural lattice-based reranker. We can categorize the retrieval algorithms into two: token-based and neural-based. The latter represent algorithms that jointly learn embeddings of queries and documents in the same embedding space and use an inner product or cosine distance to measure the similarity between queries and documents. Very recent examples of these type of methods are ColBERT [9] and ANCE [32]. However, although these techniques have proven to be very

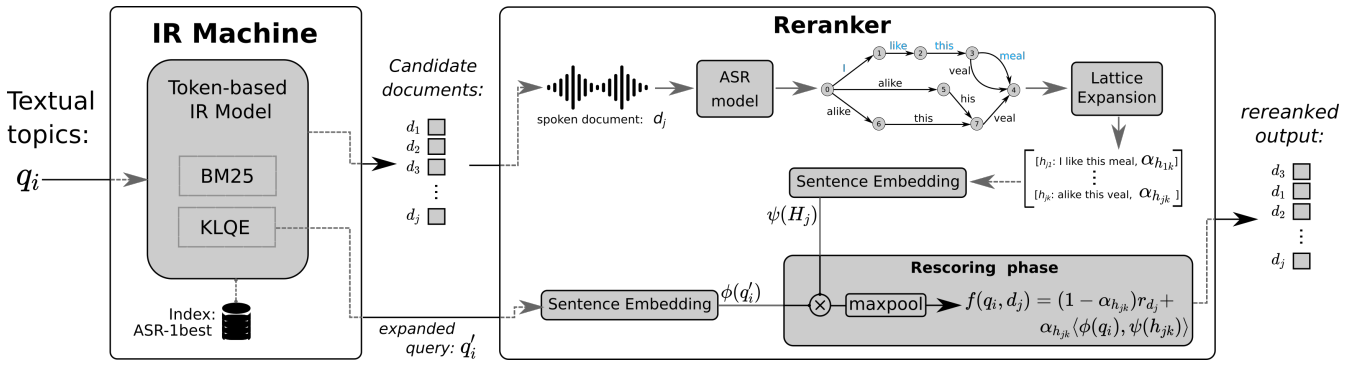

Figure 1: General overview of the proposed multi-stage retrieval architecture based on expanded lattice embeddings.

efficient in many text document retrieval tasks, they require large amounts of data, i.e., pairs (topics,relevant documents) to accurately learn the joint representation. On the other hand, the classic information retrieval algorithms (token-based) have the advantage of not requiring any exhaustive training, and have proven to be very competitive in many IR tasks. For example, the BM25 algorithm [28] is frequently used as a strong baseline in many TREC competitions. During our experiments we use the BM25 method as our first stage retrieval combined with the KL query expansion technique [2], a divergence from randomness query expansion model based on Kullback Leibler divergence that serves to rewrite the query based on the occurrences of terms in the feedback documents provided for each query.

The second stage of the proposed retrieval methodology involves an ad-hoc reranking step, i.e., documents D^+ are ranked for a given query q_i according to a relevance estimate [31]. In particular, the proposed ranking methodology calculates the retrieval score $f(q_i, d_j)$ using similarities within an expanded lattice embedding space. Although there is plenty of research exploring the advantages of contextualized embeddings (e.g., BERT) in text retrieval and ranking [14, 19, 26, 34], our work explores the impact of these representation models in combination with lattice expansion techniques to improve the performance of an SDR system.

Thus, once an initial set of documents is retrieved by the first stage retrieval, the input to the neural reranker is the corresponding audio file of $d_j \in D^+$. The lattice representation of d_j is generated and fed to the lattice expansion module (see Section 2), resulting in the minimum set of hypotheses $h_{jk} \in H_j$ for document d_j with its corresponding confidence scores $\alpha_{h_{jk}}$. Next, query q_i and hypotheses H_j are encoded through functions $\phi : Q \rightarrow \mathbb{R}^m$ and $\psi : H \rightarrow \mathbb{R}^m$ which map a sequence of tokens to their associated sentence embeddings $\phi(q)$ and $\psi(h)$, respectively. A max-pool operator is applied to the output of the inner product between $\phi(q_i)$ and the expanded lattice embeddings $\psi(H_j)$. Then, the final scoring function ($f : \mathbb{R}^m \times \mathbb{R}^m \rightarrow \mathbb{R}$) is defined as follows:

$$f(q_i, d_j) = (1 - \alpha_{h_{jk}}) r_{d_j} + \alpha_{h_{jk}} \langle \phi(q_i), \psi(h_{jk}) \rangle \quad (1)$$

where r_{d_j} is the score assigned to document d_j by the base retrieval model, $\langle \phi(q_i), \psi(h_{jk}) \rangle$ is the maximum similarity score found in the expanded lattice embedding space, and $\alpha_{h_{jk}}$ corresponds to the confidence score assigned by the lattice expansion module to hypothesis h_{jk}. Thus, our model rewards documents with good scores form the base retrieval model and high ASR confidence, and penalizes documents with low ASR confidence. By following this approach, we are searching among the expanded lattice embedding space for alternative hypotheses with the highest semantic similarity to the input query. We will refer to this method as EL-Viterbi in our experiments.

4 EXPERIMENTAL SETUP

4.1 AMI search collection

We evaluate our proposed approach on a generic meeting corpus, the AMI meetings corpus. The AMI corpus [4] is a collection of 171 meting records where groups of people are engaged in a 'role play' as a team and each speaker assumes a certain role in a team (e.g., project manager). The entire corpus represents around 100h of annotated data. Every meeting lasts about 30 minutes each, involves up to 4 speakers, covering a number of topical areas with variation of speech delivery styles. Recordings were made using 6 cameras and 12 microphones: one headset microphone for each speaker.

For performing our experiments we applied the same setup as [5] in the construction of the search collection. We merged the per speaker transcripts using the time marking data provided in the corpus to form a single transcript file for each meeting. Then, we applied a time-based segmentation, with segment boundaries placed at regular intervals of 180s with no overlap.[1] Time-based segmentation is a common procedure when facing a spoken document retrieval task, especially since its assumed that a user wants to find relevant segments from the audios without listen to them in their entirety or even to read through a transcript file [8]. After the segmentation process, our final AMI search collection is composed by 2048 segments, i.e., spoken documents.

Similarly, topics were constructed and split as described in [5], i.e., using 35 of the PowerPoint slides provided with the AMI corpus. We only report results on the test partition of the search topics (25 topics).[2] On average, each topic has 49 relevant audio segments, the topic with the minimum number of relevant documents has

[1]The decision of creating segments of 180s was made based on the observations reported in the work by [5]

[2]Search topics with their corresponding relevance assessments are available here: https://github.com/villatoroe/AMI_SDR_Queries

only 1 relevant segment, while the topic with the highest number of relevant documents has 118.

4.2 Base retrieval

For the implementation of our base IR system we used the PyTerrier platform [15]. We indexed the spoken document collection using the generated transcriptions from our one-best ASR model (see Section 5). Topics are expanded with the 40 most relevant terms from the top 3 retrieved documents using the KLQE technique.

Results are reported in terms of mean average precision (MAP) and normalized discounted cumulative gain (NDCG) at a cut-off rank (rc) of 50 and 1000.

4.3 Sentence Embeddings

We evaluated two different approaches to obtain the sentence embeddings, namely: sentence transformers (SBERT) [27], and RoBERTa [13]. For experiments using SBERT, we employed the pretrained MSMARCO passage model [7].[3] For experiments using the RoBERTa model, we re-trained the language model to better capture the language characteristics of conversational data. For this, we trained the `roberta-based`[4] model for 25 epochs using the Fisher [1] dataset (henceforth, 'R-Fisher'), a conversational speech dataset. We consider the [CLS] token as the sentence embedding.

4.4 Experiments definition

Gold - corresponds to IR performance when the retrieval is done using ground truth (i.e., error-free) transcripts. ***Oracle*** - maximum achievable results from the base IR model. **One-best** - the top-scoring ASR hypothesis is fed to the rescoring phase (Fig.1). For this setup, h_{j1} gets an $\alpha_{h_{j1}} = 0.5$, as its the most salient (confident) output of the ASR. ***n*-best** - the top n-scoring paths in the lattice ($n = 100$) are fed to the rescoring stage. Confidence scores $\alpha_{h_{jk}}$ are obtained by combining the AM and LM scores in the log domain. The AM score scaled by an acoustic scale factor of 0.1 (to match scale of the two scores) is added to the LM score. **EL-Viterbi** - we apply the constrained minimum path cover algorithm. Confidence scores $\alpha_{h_{jk}}$ are extracted from the Viterbi Forward-Backward costs that are used to compute the minimum path cover output described in [10]. **EL-AM-LM** - this configuration is a variant of EL-Viterbi where we use the AM and LM scores as in the case of n-best instead of the Viterbi Forward-Backward scores. In other words, for each path returned by the lattice expansion algorithm, we compute the AM and LM scores, and combine them similar to the n-best setup.

In the latter three experiments, a softmax operation is applied over the combined scores of all hypotheses in an utterance (i.e., d_j) to normalize their values between 0 and 1. Resulting values are considered as the confidence scores $\alpha_{h_{jk}}$.

Following methods serve as comparative evaluations: **COLBERT** [9, 16] - we use the checkpoint trained by the University of Glasgow on the MSMARCO passage ranking dataset.[5] **ESearch** - an implementation of a SBERT-based dense retrieval machine using the low-level Python client for Elasticsearch.[6] **ANCE** [32] - we used the model checkpoint listed on the ANCE github repository.[7] And, **Eskevich** [5] - reported performance by M. Eskevich and G. F. Jones in a similar SDR setup, using the provided transcripts within the AMI dataset.

[3]https://huggingface.co/sentence-transformers/msmarco-distilbert-base-v2
[4]https://huggingface.co/roberta-base
[5]http://www.dcs.gla.ac.uk/~craigm/ecir2021-tutorial/colbert_model_checkpoint.zip
[6]https://www.elastic.co/guide/en/elasticsearch/client/python-api/current/

Table 1: WERs on the AMI dataset. AMI (full): the entire collection of audio documents in AMI, AMI (eval): the standard evaluation set, and AMI (train): the training set.

System	Training data	AMI (full)	AMI (eval)
TDNN-F	Librispeech (960h)	45.5	44.6
TDNN-F [24]	AMI (train)	-	23.9

5 RESULTS AND ANALYSIS

5.1 ASR results

We trained the acoustic model with 960h of the Librispeech dataset [20]. We implemented the factorized Time-delay Neural Network (TDNN) [23] in Pytorch [21] (available with Pkwrap [17]) and followed the standard pipeline for training a Lattice Free-Maximum Mutual Information (LF-MMI) [22] based ASR typically found in Kaldi [24]. To evaluate the ASR on the AMI dataset, we used a LM trained with the transcripts from the Fisher corpus [1]. This is done for the following two reasons: (1) the Fisher corpus is used as a base LM for interpolation in the Kaldi recipe for AMI, and (2) the speech style of the Fisher dataset suits AMI dataset better than that of Librispeech.

WERs on the entire AMI dataset and the standard AMI evaluation set are presented in Table 1. The latter is presented to demonstrate the effect of not using any domain-specific data to train the AM and the LM. Specifically, we observe an absolute degradation of 20.7% in WER when not using AMI train data.

5.2 IR results

Retrieval results are shown in Table 2. On the one hand, results indicate that if we apply our neural reranker to the first 50 retrieved documents (cut-off=50), the performance of the IR machine remains more or less the same among the different alternatives to expand the lattice, i.e., n-best, EL-Viterb, and EL-AM-LM. Nevertheless, these results are significantly better than those obtained by the one-best configuration. On the other hand, when we set the cut-off=1000, the best performance is obtained when we use the minimum set of hypotheses extracted by the EL-AM-LM approach. Although EL-AM-LM and EL-Viterbi use the same approach to extract the minimum set of hypotheses from the lattice, the way the confidence score ($\alpha_{h_{jk}}$) is computed varies (see section 4.4). Thus, the way the AM and the LM scores are combined plays an important role in the definition of the hypotheses' confidence values.

Figure 2 shows the precision-recall curves of the *Gold*, n-best and the EL-AM-LM configurations. As can be observed, the EL-AM-LM model performs consistently well across several recall values.

[7]https://github.com/microsoft/ANCE/#results

Table 2: Retrieval results with cut-off at rank 50 and 1000. Symbol ‡ indicates statistical significant results ($P = 0.05$) against the n-best configuration.

IR system configuration	Encoding	*(cut-off=50)* MAP	*(cut-off=50)* NDCG	*(cut-off=1000)* MAP	*(cut-off=1000)* NDCG
Gold	n/a	0.334	0.519	0.445	0.753
Oracle	n/a	0.460	0.583	0.914	0.940
One-best	SBERT	0.211	0.412	0.066	0.439
n-best	SBERT	0.300	0.480	0.357	0.672
	R-Fisher	0.305	0.484	0.374	0.691
EL-Viterbi	SBERT	0.301	0.483	0.350	0.675
	R-Fisher	0.302	0.483	0.356	0.679
EL-AM-LM	SBERT	0.301	0.482	**0.381**‡	**0.695**‡
	R-Fisher	0.303	0.484	**0.399**‡	**0.714**‡
COLBERT	n/a	-	-	0.212	0.603
E-SEARCH	SBERT	-	-	0.229	0.574
ANCE	n/a	-	-	0.088	0.248
Eskevich	n/a	0.360	-	-	-

5.3 Ablation study

For the following experiments (Table 3), we assume that the most similar hypothesis $h_{j,k}$ will always have a high confidence score by manually fixing the value of $\alpha_{h_{jk}}$. This study shows that such a hypothesis not necessarily is the correct one, negatively affecting the reranking process.

Table 3: Retrieval results with cut-off at 1000, and $\alpha_{h_{jk}} = 0.9$

IR conf	Encoding	MAP	NDCG
n-best	R-Fisher	0.261	0.643
EL-Viterbi	R-Fisher	0.258	0.640
EL-AM-LM	R-Fisher	0.255	0.637

6 CONCLUSIONS

This paper proposed a neural reranking method based on the ASR expanded lattice embedding space. Our approach uses the base retrieval model, to obtain the first set of candidate documents. Then, in the reranking stage, a lattice expansion approach in combination with sentence embedding techniques allows searching for alternative (semantically relevant) hypotheses in the lattice embedding space. The obtained results validate the existence of richer ASR hypotheses in the lattice, which help improve the performance of the SDR system. We showed the impact of how the ASR confidence score can affect the rescoring function. The reported results define a new baseline for AMI SDR systems using ASR models trained with out-of domain data, allowing us the possibility to apply this technique in other domains (domain-agnostic).

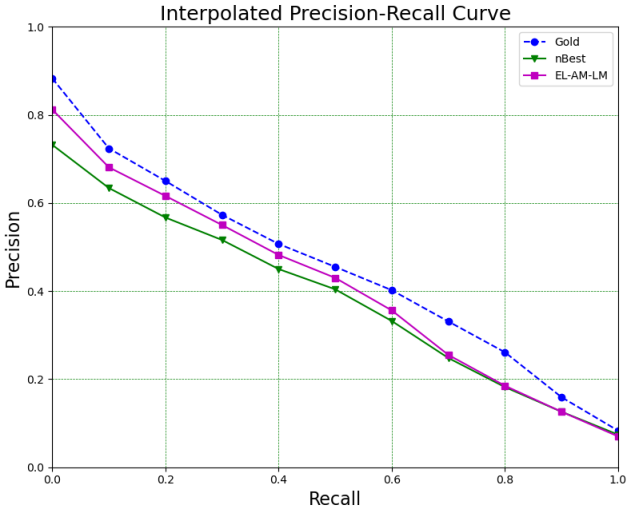

Figure 2: Precision-Recall curves at rank cut-off = 1000.

ACKNOWLEDGMENTS

Esaú Villatoro-Tello, was supported partially by Idiap, SNI CONACyT, and UAM-Cuajimalpa Mexico. The research was also partially based on the work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via AFRL Contract #FA8650-17-C-9116. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The authors would like to thank Maria Eskevich, and Gareth F. Jones, for their support in providing the AMI topics and relevance assessments that allowed us to replicate the setup described in their paper [5]. Additionally, the authors would like to acknowledge the work of Debasis Ganguly, Wei Li, Agnes Gyarmati, and Jinming Min for their assistance with manual relevance assessment.

REFERENCES

[1] 2004. Fisher English Training Speech Part 1 Transcripts LDC2004T19. (2004).
[2] Gianni Amati and Cornelis Joost Van Rijsbergen. 2002. Probabilistic Models of Information Retrieval Based on Measuring the Divergence from Randomness. *ACM Trans. Inf. Syst.* 20, 4 (oct 2002), 357–389. https://doi.org/10.1145/582415.582416
[3] Elizabeth Boschee et al. 2019. SARAL: A low-resource cross-lingual domain-focused information retrieval system for effective rapid document triage. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. 19–24.
[4] Jean Carletta. 2007. Unleashing the killer corpus: experiences in creating the multi-everything AMI Meeting Corpus. *Language Resources and Evaluation* 41, 2 (2007), 181–190.
[5] Maria Eskevich and Gareth J.F. Jones. 2014. Exploring speech retrieval from meetings using the AMI corpus. *Computer Speech & Language* 28, 5 (2014), 1021–1044. https://doi.org/10.1016/j.csl.2013.12.005
[6] Parisa Haghani et al. 2018. From audio to semantics: Approaches to end-to-end spoken language understanding. In *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 720–726.
[7] Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently Teaching an Effective Dense Retriever with Balanced

Topic Aware Sampling. In *Proc. of SIGIR*.

[8] Rosie Jones, Ben Carterette, Ann Clifton, Maria Eskevich, Gareth JF Jones, Jussi Karlgren, Aasish Pappu, Sravana Reddy, and Yongze Yu. 2021. Trec 2020 podcasts track overview. *arXiv preprint arXiv:2103.15953* (2021).

[9] Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, China) *(SIGIR '20)*. Association for Computing Machinery, New York, NY, USA, 39–48. https://doi.org/10.1145/3397271.3401075

[10] Ke Li, Daniel Povey, and Sanjeev Khudanpur. 2021. A parallelizable lattice rescoring strategy with neural language models. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 6518–6522.

[11] Xunying Liu, Xie Chen, Yongqiang Wang, Mark JF Gales, and Philip C Woodland. 2016. Two efficient lattice rescoring methods using recurrent neural network language models. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 24, 8 (2016), 1438–1449.

[12] Xunying Liu, Yongqiang Wang, Xie Chen, Mark JF Gales, and Philip C Woodland. 2014. Efficient lattice rescoring using recurrent neural network language models. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 4908–4912.

[13] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692 [cs.CL]

[14] Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. 2019. CEDR: Contextualized Embeddings for Document Ranking. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Paris, France) *(SIGIR'19)*. Association for Computing Machinery, New York, NY, USA, 1101–1104. https://doi.org/10.1145/3331184.3331317

[15] Craig Macdonald and Nicola Tonellotto. 2020. Declarative Experimentation inInformation Retrieval using PyTerrier. In *Proceedings of ICTIR 2020*.

[16] Craig Macdonald and Nicola Tonellotto. 2021. On Approximate Nearest Neighbour Selection for Multi-Stage Dense Retrieval. *Proceedings of the 30th ACM International Conference on Information & Knowledge Management* (Oct 2021). https://doi.org/10.1145/3459637.3482156

[17] Srikanth Madikeri et al. 2020. Pkwrap: a pytorch package for lf-mmi training of acoustic models. *arXiv preprint arXiv:2010.03466* (2020).

[18] Lidia Mangu, Eric Brill, and Andreas Stolcke. 2000. Finding consensus in speech recognition: word error minimization and other applications of confusion networks. *Computer Speech & Language* 14, 4 (2000), 373–400.

[19] Rodrigo Nogueira and Kyunghyun Cho. 2020. Passage Re-ranking with BERT. arXiv:1901.04085 [cs.IR]

[20] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 5206–5210.

[21] Adam Paszke et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019), 8026–8037.

[22] Daniel Povey et al. 2016. Purely sequence-trained neural networks for ASR based on lattice-free MMI.. In *Interspeech*. 2751–2755.

[23] Daniel Povey et al. 2018. Semi-Orthogonal Low-Rank Matrix Factorization for Deep Neural Networks.. In *Interspeech*. 3743–3747.

[24] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. 2011. The Kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society.

[25] Daniel Povey, Mirko Hannemann, Gilles Boulianne, Lukáš Burget, Arnab Ghoshal, Miloš Janda, Martin Karafiát, Stefan Kombrink, Petr Motlíček, Yanmin Qian, et al. 2012. Generating exact lattices in the WFST framework. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 4213–4216.

[26] Yifan Qiao, Chenyan Xiong, Zhenghao Liu, and Zhiyuan Liu. 2019. Understanding the Behaviors of BERT in Ranking. arXiv:1904.07531 [cs.IR]

[27] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. https://arxiv.org/abs/1908.10084

[28] Stephen Robertson and Hugo Zaragoza. 2009. *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc.

[29] Nicholas Ruiz, Mattia Antonino Di Gangi, Nicola Bertoldi, and Marcello Federico. 2017. Assessing the Tolerance of Neural Machine Translation Systems Against Speech Recognition Errors. *Interspeech 2017* (Aug 2017). https://doi.org/10.21437/interspeech.2017-1690

[30] Ville T Turunen and Mikko Kurimo. 2007. Indexing confusion networks for morph-based spoken document retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. 631–638.

[31] Esaú Villatoro-Tello, Antonio Juárez-González, Manuel Montes-y Gómez, Luis Villaseñor-Pineda, and L. Enrique Sucar. 2012. Document ranking refinement using a Markov random field model. *Natural Language Engineering* 18, 2 (2012), 155–185. https://doi.org/10.1017/S1351324912000010

[32] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. arXiv:2007.00808 [cs.IR]

[33] Hainan Xu, Tongfei Chen, Dongji Gao, Yiming Wang, Ke Li, Nagendra Goel, Yishay Carmiel, Daniel Povey, and Sanjeev Khudanpur. 2018. A pruned rnnlm lattice-rescoring algorithm for automatic speech recognition. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 5929–5933.

[34] Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Simple Applications of BERT for Ad Hoc Document Retrieval. arXiv:1903.10972 [cs.IR]

[35] Le Zhang et al. 2020. The 2019 bbn cross-lingual information retrieval system. In *Proceedings of the workshop on Cross-Language Search and Summarization of Text and Speech (CLSSTS2020)*. 44–51.