Performance of Ad-hoc Retrieval with Automatic Speech Recognition Transcribed Queries

Max Robinson max.robinson@jhu.edu
Johns Hopkins University,
Baltimore, MD 21218 USA

Abstract

Ad-hoc retrieval with automatic speech recognition (ASR) transcribed queries presents a new set of challenges. The types of errors in transcribed queries are not the same as queries that were typed. This paper looks at the FIRE 2010 dataset with transcribed queries and investigates which simple indexing schemes provide the most benefit to mitigate the errors that arise from ASR. The results showed that while the performance of ad-hoc retrieval is significantly degraded by poor transcription, using an indexing type of character 4-grams provided the best mitigation for errors introduced by ASR compared to character 3-grams and word indexing.

1. Introduction

Ad-hoc retrieval is a staple in modern day technology. With the world wide web and services such as Google and Microsoft's Bing, searching for relevant information has common practice. Now as technology has continued to advance, automatic speech recognition is again at the forefront of technology through voice user interfaces for information retrieval.

Much of where ASR is used today is to enable easier search for information. Many smart devices, including smart phones, have some sort of on board or built in ASR that is used for search. Everything from searching the Internet to looking up movies can have integration with ASR for these searches.

The goal of ASR is to have perfect accuracy in transcription of what a user says to what the system outputs as text. The ability accurately transcribe text in an every day scenario similar to a human has shown to be a difficult task. This provides an interesting problem to information retrieval systems to still provide relevant information even though the query could have errors.

Providing accurate document retrieval with errors like spelling mistakes in queries is an active area of research. However, ASR transcribed text presents a new problem with the possibility of a transcribed word being entirely different than the intended word by the speaker. While, more accurate ASR would help fix this problem, it is not clear that all ASR will reach near perfect accuracy soon.

This paper explores how different indexing types for ad-hoc document retrieval affect the performance retrieval using queries that are transcribed with ASR. To do so, this paper uses the English FIRE 2010 corpra (https://www.isical.ac.in/ clia/2010/) to test character 3-gram, 4-gram, and word based tokenization and indexing. For retrieval and executing queries, the open source search engine Elasticsearch is used. Mozilla's DeepSpeech imple-

mentation (Mozilla, 2014) was used to transcribe audio versions of the FIRE 2010 queries used for testing.

2. Previous Work and Background

Automatic speech recognition saw research into the domain with working systems as early as 1952 with a single speaker digit recognition system built by Bell Labs (Juang & Rabiner, 2005). Since then, ASR has come a long ways. Services from companies like Google and Micrsoft are reaching close to 95% accuracy (Meeker, 2018) (Xiong, Wu, Alleva, Droppo, Huang, & Stolcke, 2017). Human accuracy is considered to be about 95% (Xiong et al., 2017) as well, meaning that ASR is reaching a human level of accuracy very quickly.

However, not all ASR is this accurate and not all uses of ASR can use tools provided by companies like Google or Microsoft. Alternative non-proprietary based solutions exist for this reason. A stand alone recurrent neural network called Deep Speech was developed by researchers at Baidu in 2014 (Hannun, Case, Casper, Catanzaro, Diamos, Elsen, Prenger, Satheesh, Sengupta, Coates, & Ng, 2014). The network is an end-to-end audio to text network, that uses a 1-best word prediction.

The researchers claimed a 16% error rate on a widely studied dataset, Switchboard Hub5'00. An open source implementation of Deep Speech by Mozilla (Mozilla, 2014) was later created with publicly available language models. This enables ASR without the need to use commercial services.

Errors in queries for information retrieval systems have also been studied. In a typical query system, errors in the query are typically caused by a human mistyping a word resulting in a spelling mistake. Spelling mistakes have been studied for sometime and there are existing solutions such as edit distance, context sensitive spelling corrections, or phonetic corrections shown in books like Manning's "Introduction to Information Retrieval" (Manning, Raghavan, & Schtze, 2008).

Other techniques such as relevance feedback and query expansion have also been used to combat non-specific queries and enhance retrieval performance. Query expansion attempts to add relevant words to a query in the hopes that the expanded query provides better results. Relevance feedback aims to adjust a query based on the top documents initially retrieved to improve results. Relevance feedback can also be used as a corrective measure to suggest to a user a more likely query if a mistake was made.

3. Experiment

To test the effect of different indexing methods on retrieval performance for ASR transcribed queries the Forum for Information Retrieval Evaluation (FIRE) 2010 (https://www.isical.ac.in/~clia/2010/) English dataset was selected. The FIRE 2010 dataset is a collection of documents focused on news articles in the South Asia region. The documents are available in a few different languages, but for the purpose of this experiment the English version of these documents will be used.

FIRE 2010 follows the TREC standard convention for document, topic, and qrels encodings. This allows the application of the standard TREC_EVAL calculations and metrics to be used. The primary metric used in this experiment is the mean average precision

(MAP) for each set of queries. The full TREC_EVAL output is available in Appendix C for reference.

In order to test ASR transcription, the queries for FIRE 2010 had to be transformed into audio and then the audio must have ASR applied to it to obtain a transcribed query. To transform the queries, the queries were spoken out loud and recorded with an inexpensive headset microphone. The audio was recorded at 16KHz with 1 channel and a chunk size of 1024.

Each recording was 20 seconds in length, even if speaking the query did not take 20 seconds. Queries were spoken with roughly the same cadence and speed to as much of a degree as possible by the speaker. The speaker was a native English speaker and words were spoken close to dictionary pronunciations typically (i.e. little to know accent). The speaker was not familiar with many of the named entities in the query text so pronunciations of non-English words were done to the best of the speakers ability, following English phonetic rules.

Mozilla's DeepSpeech implementation was chosen to transcribe the audio queries back to text. Mozilla's DeepSpeech was chosen because of its free availability and its popularity in the open source community. In addition, DeepSpeech is free and able to be run on a single computer with middling specifications. The model used for DeepSpeech is the freely available v0.3.0 model retrievable via the Mozilla DeepSpeech repository.

Elasticsearch (www.elastic.co) was used as the search engine and document storage. Elasticsearch is an open source and commercial search engine built on top of Apache Lucene (lucene.apache.org) which is an open source text search engine written in Java. Elasticsearch allows documents to be placed into different indexes and each index can specify a mapping, text analyzer, and tokenizer for indexing and querying on specific fields.

Three indexes were created and used for the experiment. One index specified characters to be indexed using character 3-grams, another using character 4-grams, and the last used the standard Elasticsearch word indexing with stopword removal. For all indexes, text was translated into all lowercase characters prior to being indexed. An example of the Elasticsearch indexing settings can be seen in Appendix A.

The documents for FIRE 2010 were indexed into Elasticsearch using a companion tool Logstash (www.elastic.co/products/logstash), after first being translated into tsv files. Logstash was used primarily to quickly index data into Elasticsearch. For each document, the entirety of the document was indexed, and the document ID was also stored with the document.

Queries were made to Elasticsearch using both the transcribed queries and the original queries for comparison. The query used was a standard Elasticsearch "match" query. Elasticsearch preprocessed the query the same way as if indexing a document according to that index, and then executes a boolean "or" query. The format of the query can be seen in Appendix B. Only the document ID was retrieved for each query and the score for each document was captured.

In an attempt to retrieve as many relevant documents as possible for each query, the top 1000 documents were retrieved for each query. There were 50 queries total and the top 1000 documents were retrieved for each of those, resulting in a total of 50,000 documents retrieved.

Figure 1: Ground Truth and Hypothesized Queries from DeepSpeech

Q77 GT: *Attacks by Hezbollah guerrillas Attacks by Hezbollah guerrillas on Indian and Israeli forces.*
Q77 Transcribed: *attack by has below garillas attacked by hasbelocarillos on indian and is rally forcees*

Q121 GT: *Blasts on Samjhauta Express Deadly explosions on the Samjhauta Express.*
Q121 Transcribed: *blasts on some shot to it press that ly explosions on these seven shot a expresss*

The results of each query were output in the standard TREC evaluation format for a trec_top_file. TREC_EVAL was run on the results from executing all 50 transcribed queries against all indexes and executing all 50 original queries against all indexes.

4. Results and Analysis

ASR transcriptions are not without error. To understand how errors can impact querying the Word Error Rate (WER) was calculated for each query. Word error rate is a metric in ASR to try and measure how good the transcription is. It is similar to the Levenshtein distance for words, but rather uses words in a sentence rather than characters in a word. WER can be calculated by $WER = \frac{S+D+I}{N}$ where S is the number of substitutions, D is the number of deletions, I is the number of inserts, and N is the number of words in the reference. Lower is better.

In addition the word recognition rate was also calculated. Word recognition rate is the number of matched words divided by the number of words in the reference sentence or document. Higher is better. The results were then averaged to determine an average word error rate across all queries. Both WER and WRR were calculated using a open source third-party python library, asr-evalutation (https://github.com/belambert/asr-evaluation).

Table 1 shows the WER and WRR for each query. Table 2 shows some aggregated statistics about the WER and WRR. In general, the WER was very high for all queries. Some queries even had WERs that were above 100%. This means is that the transcribed text required more edits ans substitutions than words in the original query. The best WER achieved was about 40%, which is still a ways off of 16% error rate claimed by Baidu researchers.

Investigating the translated queries leads to a hypothesis for the poor ASR performance. While the FIRE 2010 dataset is a set of English language documents and queries, many of the named entities in the queries are not English words. However, the model used for DeepSpeech was an English model. These non-English words might not have been covered by the training or test set for the model. This could cause many errors in query translation.

Figure 1 shows two example ground truth queries from FIRE 2010 and the counter part transcription from DeepSpeech. In Q77, “Hezbollah” can be seen to cause a failure in transcription just for that word, and thus loosing the named entity in the sentence. Q121, the word “Samjhauta” appears to cause a failure in transcription but not only for

Table 1: Word Error Rate and Word Recognition Rage Per Query

Query ID	WER %	WRR %
76	55	55
77	91.667	25
78	103.704	44.444
79	62.5	50
80	80	45
81	55.556	55.556
82	85.714	57.143
83	120.833	29.167
84	72.222	50
85	78.571	28.571
86	62.5	54.167
87	87.097	19.355
88	68.571	31.429
89	92.308	38.462
90	70.833	50
91	84.615	46.154
92	75	25
93	40	70
94	47.368	63.158
95	63.158	52.632
96	100	23.077
97	66.667	47.619
98	92.857	42.857
99	106.667	33.333
100	85.714	42.857

Query ID	WER %	WRR %
101	77.778	48.148
102	86.667	53.333
103	100	47.368
104	95.833	37.5
105	68	52
106	93.33	6.667
107	89.474	52.632
108	89.474	36.842
109	43.478	65.217
110	107.143	42.857
111	66.667	53.333
112	94.118	41.176
113	56.25	62.5
114	67.857	53.571
115	116.667	41.667
116	88.235	47.059
117	50	67.857
118	94.444	44.4444
119	100	0
120	86.207	13.793
121	144.44	33.333
122	100	35.714
123	111.111	22.222
124	50	62.5
125	85.714	42.857

Table 2: Aggregated WER and WRR percentages

	Average %	Min %	Max %
WER	82.24	40	144.44
WRR	42.891428	0	70

the single word. In this case, many following words seem to be effected by this mistake in transcription.

In addition to non-English words in the queries, the speaker who was recorded was also unfamiliar with the pronunciation of many of these non-English words. The resulting audio for these words could also be poor as a result as the pronunciation could be incorrect which could lead to possibly worse results. The audio quality for query transcription could also have played a factor as additional noise from the mic or background could have played a role. The audio recording setup was not dislike many setups for a consumer, so the audio quality should be at least partially representative of a real world circumstance.

Table 3: % of Ground Truth MAP per Indexing Type

Indexing	Trans	GT	% of GT performance
3-gram	0.1435	0.4162	34.47861605
4-gram	0.2055	0.4516	45.50487157
words	0.1302	0.3789	34.36262866

The effect ASR error had on query results was substantial. Figure 2 shows the MAP for queries using transcribed queries versus the ground truth queries across all indexing types; 3-grams, 4-grams, and words. It can be seen that transcribed queries did significantly worse than their ground truth counter parts in all indexing schemes.

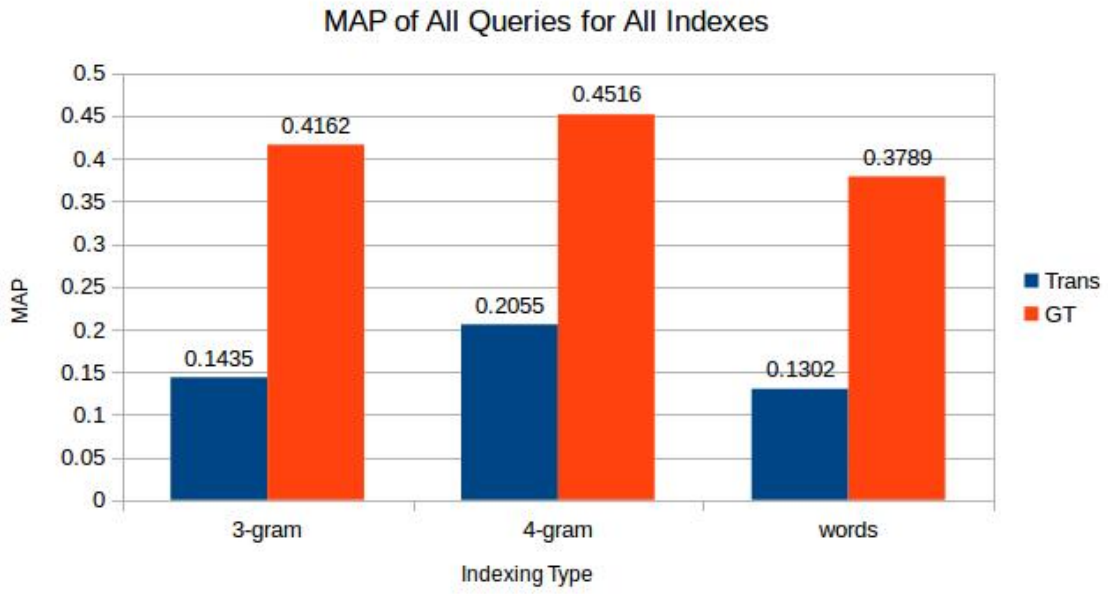

Figure 2: MAP of Ground Truth and Transcribed Queries for All Indexing Schemes

Overall, 4-gram indexing had the best MAP both for the ground truth queries and for transcribed queries. This provides additional evidence to previous works that showed that 4-grams are particularly effective, especially for European languages (McNamee & Mayfield, 2004). In addition, previous work showed that 4-grams are the most effective for the Hindi FIRE 2010 dataset (Vishwakarma, Lakhtaria, Bhatnagar, & Sharma, 2015). This seems to be the case for the English FIRE 2010 dataset.

While the performance of 4-grams has been shown before, the interesting part for this experiment is how 4-grams were able to reduce the impact that ASR error had on query performance. Table 3 shows the percentage of performance the transcribed query had compared to its ground truth counter. The transcribed queries using 3-grams performed only 34% as well when compared to ground truth queries using 3-grams. So transcription decreased MAP by about two-thirds for 3-grams. The same is true for words, about a two-thirds reduction in performance.

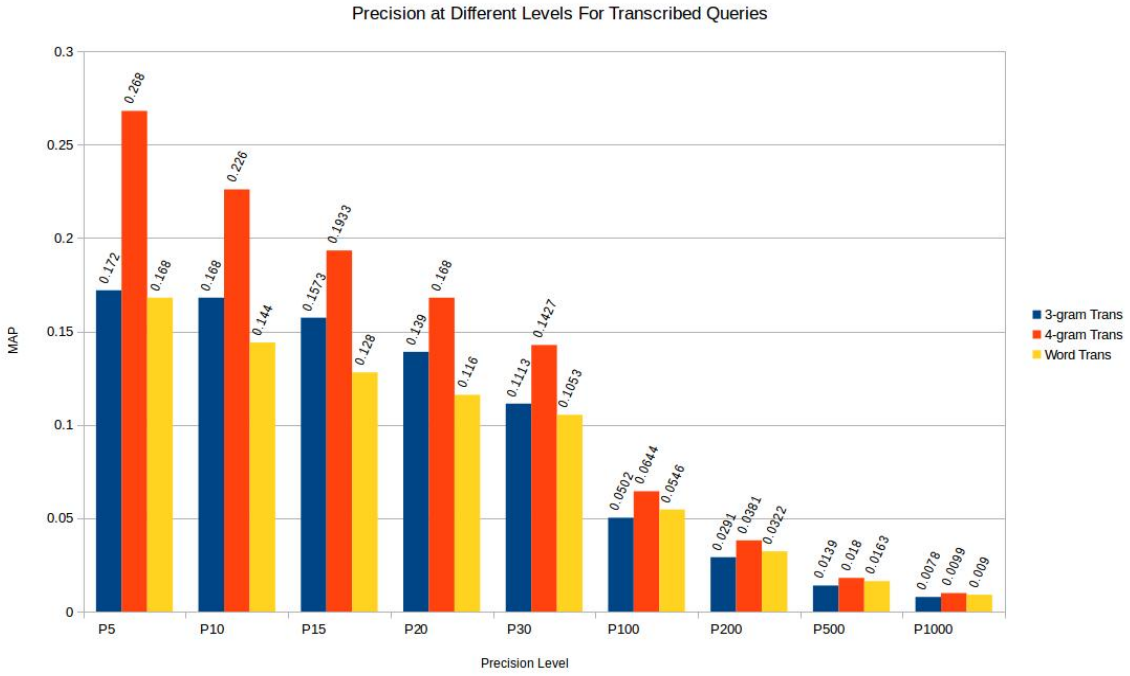

Figure 3: Precision at different Levels for Transcribed Queries

For 4-grams, only had about a 55% reduction in performance compared to its ground truth counter parts. This shows that 4-grams were more effective at reducing the effect of ASR error on retrieval than 3-grams and words.

The performance of 4-grams can also be seen when comparing precision with different numbers of documents retrieved. Figure 3 shows precision for the number of documents retrieved. 4-grams can be seen to have much higher precision compared to 3-grams and words especially when fewer documents are retrieved. This indicates that 4-grams would also perform better if only looking at the top 30 or so documents.

5. Future Work

While this paper explored some aspects of ASR integration with ad-hoc retrieval there is much more work that could be done. The DeepSpeech off the shelf model for ASR seems to need a different dataset for queries if the effects of ASR with document retrieval are to be inspected with a lower average word error rate. The high word error rate provided good contrast for this paper, more realistic performance might be achieved by using a dataset consisting of common English words to transcribe.

More work can also be done to analyze where ASR makes mistakes and how other forms of tokenization and indexing can possibly reconcile these errors. Stemming for transcriptions would be an area of future work to possibly avoid plural or past tense errors. Soundex type indexing might be tried in order to capture the sounds of transcribed words that might account for errors.

6. Conclusion

In this paper, the overlap between automatic speech recognition and ad-hoc retrieval was explored. The interesting problem of errors through transcriptions motivated a look what basic tokenizing and indexing techniques could be applied to mitigate these errors. While the ASR used for the experiments had high word error rates for transcribed queries, it magnified the differences between the different indexing approaches.

The results of the experiment showed that character 4-grams where the best indexing choice to mitigate error introduced by transcribing queries using the Mozilla DeepSpeech implementation. Not only did 4-grams perform mitigate error best overall, it also provided the best performance for looking at only the top documents returned.

While more work can be done to explore additional techniques to mitigate error from ASR, this work reaffirms that character 4-grams perform well in document retrieval and are good starting place for ASR transcription error mitigation for document retrieval.

References

Hannun, A. Y., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A., & Ng, A. Y. (2014). Deep speech: Scaling up end-to-end speech recognition. *CoRR*, *abs/1412.5567*.

Juang, B. H., & Rabiner, L. R. (2005). Automatic speech recognition - A brief history of the technology development. *Elsevier Encyclopedia of Language and Linguistics*.

Manning, C. D., Raghavan, P., & Schtze, H. (2008). *Introduction to Information Retrieval* (1 edition). Cambridge University Press.

McNamee, P., & Mayfield, J. (2004). Character n-gram tokenization for european language text retrieval. *Information Retrieval*, *7*(1), 73–97.

Meeker, M. (2018). Mary meeker's 2018 internet trends report. electronic. Published at Code Conference 2018.

Mozilla (2014). https://github.com/mozilla/deepspeech..

Vishwakarma, S. K., Lakhtaria, K. I., Bhatnagar, D., & Sharma, A. K. (2015). Monolingual information retrieval using terrier: Fire 2010 experiments based on n-gram indexing. *Procedia Computer Science*, *57*, 815 – 820. 3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015).

Xiong, W., Wu, L., Alleva, F., Droppo, J., Huang, X., & Stolcke, A. (2017). The microsoft 2017 conversational speech recognition system. Tech. rep..

Appendix A: Elasticsearch Index 4-gram Configuration

```
{
  "settings":{
    "number_of_shards" :   1,
    "number_of_replicas" : 0,
    "analysis":{
      "analyzer":{
        "4gram":{
           "type":"custom",
           "tokenizer":"4grams",
           "filter":[
              "lowercase"
           ]
        }
       },
      "tokenizer": {
        "4grams": {
          "type": "ngram",
          "min_gram": 4,
          "max_gram": 4,
          "token_chars": []
        }
      }
    }
  },
  "mappings": {
    "doc": {
      "properties": {
        "doc": {
          "type": "text",
          "analyzer":"4gram",
          "search_analyzer":"4gram"
        }
      }
    }
  }
}
```

Appendix B: Elasticsearch Query

```
{
    "_source": ["docID"],
    "query": {
        "match": {
            "doc": "<query text>"
        }
    },
    "size": 1000
}
```

Appendix C: Full TREC_EVAL output

	3-grams trans	3-grams gt	4-grams trans	4-grams gt	words trans	words gt
num_q	50	50	50	50	50	50
num_ret	50000	50000	50000	50000	50000	50000
num_rel	653	653	653	653	653	653
num_rel_ret	392	640	497	650	449	635
map	0.1435	0.4162	0.2055	0.4516	0.1302	0.3789
gm_ap	0.0175	0.3017	0.0583	0.339	0.0212	0.2496
R-prec	0.1581	0.3913	0.2143	0.4172	0.1316	0.3553
bpref	0.5646	0.9785	0.7407	0.9973	0.6417	0.9665
recip_rank	0.3471	0.6968	0.4959	0.735	0.2818	0.676
ircl_prn.0.00	0.3553	0.7339	0.5168	0.7625	0.3048	0.7134
ircl_prn.0.10	0.2906	0.6998	0.4051	0.6912	0.241	0.6264
ircl_prn.0.20	0.2584	0.5787	0.3192	0.6113	0.209	0.5316
ircl_prn.0.30	0.1776	0.5185	0.2624	0.5544	0.1641	0.4538
ircl_prn.0.40	0.1552	0.4633	0.2393	0.5165	0.1461	0.4055
ircl_prn.0.50	0.1313	0.4093	0.2095	0.4607	0.1272	0.3715
ircl_prn.0.60	0.1088	0.3786	0.1533	0.4267	0.1109	0.3432
ircl_prn.0.70	0.0788	0.3208	0.1245	0.3655	0.0885	0.3264
ircl_prn.0.80	0.0565	0.2797	0.1025	0.3276	0.0778	0.2741
ircl_prn.0.90	0.041	0.2256	0.0708	0.2508	0.0505	0.2061
ircl_prn.1.00	0.0359	0.1957	0.0483	0.206	0.0316	0.1592
P5	0.172	0.508	0.268	0.536	0.168	0.412
P10	0.168	0.38	0.226	0.408	0.144	0.338
P15	0.1573	0.336	0.1933	0.3573	0.128	0.3067
P20	0.139	0.297	0.168	0.318	0.116	0.267
P30	0.1113	0.2467	0.1427	0.2513	0.1053	0.2213
P100	0.0502	0.1096	0.0644	0.1164	0.0546	0.1052
P200	0.0291	0.0608	0.0381	0.0623	0.0322	0.0574
P500	0.0139	0.0252	0.018	0.0257	0.0163	0.0247
P1000	0.0078	0.0128	0.0099	0.013	0.009	0.0127