Language Models for Questions*

Edward Schofield
Telecommunications Research Center Vienna (ftw)
Vienna, Austria
schofield@ftw.at

April 2003

Abstract

Natural-language question answering is a promising interface for retrieving information in mobile contexts because it bypasses the problem of presenting documents and interim search results on a small screen. This paper considers language models suitable for rapid predictive textual and spoken input of natural-language questions. It describes a varied corpus of fact-seeking questions posed by users online and analyzes its structure. We find it to be highly constrained lexically despite its wide spectrum of topics, with a per-word perplexity less than 47 with around 2.6% of words in the test set out-of-vocabulary. One implication is that predictive interfaces can greatly speed up the input of natural-language questions with a keypad or stylus. Another is that automatic speech recognition of such questions can be quite accurate.

1 Introduction

Mobile devices have relatively small screens, and are cumbersome to use for retrieving information with traditional interfaces that return lists of matching documents in response to keywords. [1] argues that the case for question answering as an alternative interface for finding information on a small device is strong because answers to questions are more likely to fit comfortably on a small screen than arbitrary documents. Question-answering systems shift a user's burden from filtering documents for relevance to describing his or her need for information more precisely at the outset with a question rather

*Published in the Proceedings of EACL-03, Budapest, Hungary

than a string of keywords. Question answering thus demands less of the output facilities of mobile devices but more of their input facilities. This paper investigates how language models customized to questions can speed up their input.

Section 2 describes the origin and nature of the corpus of natural-language questions analyzed in Section 3, which compares various n-gram language models and shows the per-word perplexity of these questions to be lower than that of the utterances modeled in several common speech-recognition tasks. The paper then discusses the implications of this for input using keypads and styluses (Section 4) and speech (Section 5). Section 6 discusses opportunities for better models of questions.

2 The corpus of questions

I collected around 450,000 questions from various online sources—logs of the Ask Jeeves and Excite search engines, FAQFinder [2], AnswerBus [3], and test questions from the TREC question-answering track [4] from 1998 to 2001—and wrote scripts to correct common typos and spelling mistakes and to filter the corpus in the various ways in Table 1. After this massaging the corpus had 279,456 unique questions. Table 2 shows a random sample of questions from the processed corpus. It still includes spelling mistakes ('alcahol'), irregular punctuation ('advanced-screening', 'science related'), and nonsense ('How can I figure?').

Table 1: Elements removed from the corpus.

strings of fewer than 3 words
duplicate questions
requests for web sites
requests for pornography
requests for downloads
various punctuation characters

Notice that some of the questions in the corpus provide no more information than a string of keywords (*e.g.* 'Where can I find information on species in salt marshes?'). This is especially common of questions beginning 'Where can I find . . . ', which constitute about 10% of the corpus. Sometimes people want general information about a new topic or pointers to general references. The capabilities of current search engines also likely encourage users to ask

Table 2: A random sample of questions from the processed corpus.

Where can i learn how to brew alcahol?
Where can i find a list section 8 providers?
Name a female figure skater.
Where can i find images of the comic dark chyld?
What are some hotels in colorado?
What is the medical disorder arrythemia?
Where can i find information on species in salt marshes?
When was the web page for the university of michigan office of new student programs last updated?
How can i see advanced-screening of movies?
Where can i find a alternative for mescaline?
Are bankers required to get continuing education?
Where can i buy science related toys?
How can i get a pc to mac ethernet connection?
Where can i find information on the human respiratory system?
Where can i find an african recipe?
I am looking for information on the massachusetts education reform.
Where can i find a picture of kurt cuban with a gun in his mouth?
How do i remove things from my favorites list?
Where can i get the pass report for snowqualmie pass?
What is a 110 punchdown block?
What is the definition of astronomy?
How can i figure?

for links to categories of information rather than directly for the information itself.

3 n-grams of questions

This section describes and compares the fit of various n-gram models to the training corpus. We trained models for $n = 2, 3, 4$ with various sizes of lexicon and 'cutoff' thresholds for ignoring infrequent sequences. For each model we randomly constructed five partitions of the corpus, each 90% for training and 10% for testing. Table 3 reports the geometric mean perplexity on the test sets per word. We trained models with both the Good–Turing and Witten–Bell discounting schemes. Table 3 reports perplexities for the

latter, denoted type C in [5], which were 1–2% lower than for Good–Turing discounting.

'Perplexity' [6] is commonly used in speech recognition research as a measure of the goodness-of-fit of language models; a language model with a lower perplexity will usually—but not always [7]—induce fewer misrecognitions for the same task. The perplexity statistic also indicates the relative difficulty of prediction across different domains. Table 5 summarizes the perplexities of language models for various benchmark tasks in speech-recognition research. Here we adopt the standard practice of excluding from calculation any words encountered in the test set that are not in the lexicon. Note that models with small lexica have artificially low perplexity scores: models with larger lexica are penalized for their unreliable predictions about infrequent words, whereas models with smaller lexica have no power whatever to predict infrequent words, and incur no penalty. Thus the perplexity computed this way is incomparable across models with lexica of different sizes, and is imperfect as a measure of a language model's predictive power.

The vocabulary of open-domain questions is potentially as large as a language itself. We can expect that, as per Zipf's second law [8], no corpus of practical size will include all words; interfaces for textual or spoken input must be designed to accommodate omissions in the lexicon gracefully. Figure 1 shows the effect of the size of the training set on the rate of occurrence of new words in unseen questions. Extrapolating, we can predict that unseen questions in this domain are unlikely to have an out-of-vocabulary rate under about 1.5% for models trained with any practically sized corpus.

Speech recognizers have little opportunity for phonetically transcribing a word not in their lexicon, especially for languages like English with many homophones. The usual consequence is a misrecognition of the offending word and often of its neighbors. Predictive typing aids can be more forgiving. Users of a stylus or keypad can input unusual words normally, ignoring any bogus suggestions. This suggests an alternative measure to perplexity for the effectiveness of language models for predictive typing, like the expected number of keystrokes per character. Such a measure [9] would depend on implementation-specific characteristics of the interface. We describe one such implementation in progress in the next section; meanwhile we conjecture that, other factors being equal, lower-perplexity models generally imply better prediction. Hingeing upon this, the relatively low perplexities in Tables 3 and 4 bode well for the impatient questioner.

Table 3: The average cross perplexities of n-gram models on a 10% test set, with the models' sizes on disk and proportions of out-of-vocabulary (OOV) words. The size of the language models is a function of cutoff thresholds (omitted) for the minimal number of occurrences of each n-gram necessary to estimate its frequency.

Lexicon size	n	Model size (MB)	OOV rate	Perplexity
8k	2	1.0	8.1%	57.9
8k	2	1.3	8.1%	47.8
8k	2	1.9	8.1%	43.7
16k	2	1.7	5.3%	58.9
16k	3	2.0	5.3%	65.6
16k	2	2.5	5.3%	52.1
16k	3	3.4	5.3%	48.1
16k	3	7.5	5.3%	39.4
32k	2	2.5	3.6%	67.8
32k	2	3.4	3.6%	59.3
32k	3	4.3	3.6%	56.1
32k	3	8.3	3.6%	45.0
65k	2	5.1	2.6%	64.2
65k	3	10	2.6%	48.8
65k	4	18	2.6%	46.6
65k	5	28	2.6%	46.6

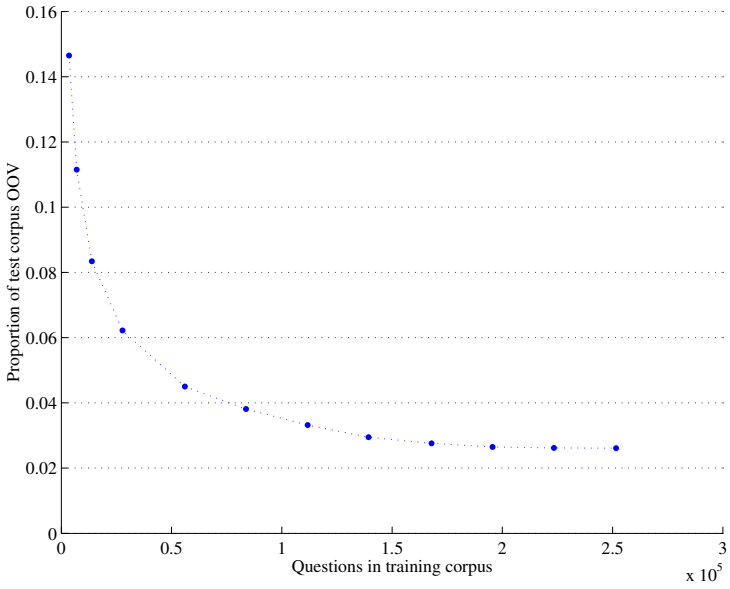
Table 4: 3-gram models clustered by initial word. PP1 is the test-set perplexity for models trained on all questions. PP2 is for models trained on those questions with the initial word.

% of corpus	Initial word	PP1	PP2	OOV
45%	Where	25.3	22.7	2.2%
22%	What	73.3	53.7	3.5%
15%	How	60.5	40.5	1.9%
4.7%	Who	111	50.1	5.3%
1.3%	When	131	–	2.3%
1.2%	Why	252	–	2.6%
1.2%	Is	207	–	3.4%
1.0%	Can	119	–	2.3%
8.7%	*Other*	362	–	4.1%

Table 5: Benchmark tasks in speech-recognition research and their approximate 3-gram language-model perplexities, with the present domain for comparison.

Task	Perplexity
Texas Instruments Digits	10
Air Travel Information System	15
Natural-language questions	*45*
Naval Resource Management	60
Switchboard	75
Wall Street Journal	170
Broadcast News	175

Figure 1: The effect of the size of the training set on the proportion of out-of-vocabulary (OOV) words.



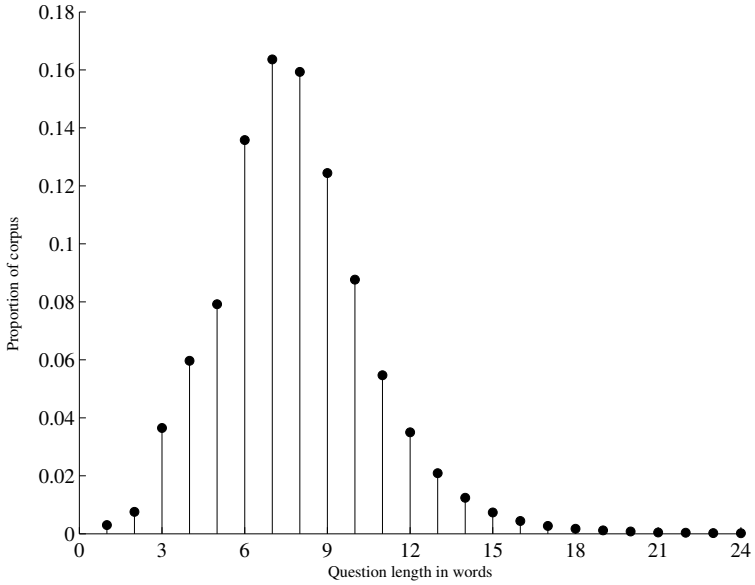


Figure 2: Breakdown of questions in the corpus by length.

4 Implications for entering questions with a keypad or stylus

Most Palm devices display about 11 lines; most Pocket PCs about 15. Locating the intended word in a list on such a screen would, with language models of perplexity 45–50, often require scrolling. So any list of likely continuations should update more frequently than once per word.

We envisage two scenarios. In the first, predictive text software would reside on the mobile device. The capacity of the device would constrain the comprehensiveness of the language models; current PDAs and handheld computers might limit them to 0.5–5 MB. For comparison, the version of T9 that Tegic Communications released in 1998 for Palm devices was 170 kB. The sizes shown in Table 3 assume four-byte floating point storage of log probabilities. Using two-byte single precision floats and a simple text compression scheme for words should roughly halve these space requirements with a negligible increase in perplexity.

The second scenario is that the device use an always-on data connection to send each chosen character to a remote server. The server would recompute its predictions with few resource constraints and return an updated list. Our tests in Austria's GPRS networks indicate that round-trip times for TCP packets are commonly around 1000 milliseconds. This may, depending on the interface, be too long to wait. We expect UMTS networks to have less latency.

We have not analyzed the average number of keystrokes or stylus gestures that an interface coupled with our language models would demand. See [9] for a comparison of the keystroke requirements of existing text-entry methods. Such a measure is more relevant than perplexity to the speed of text entry but is interface-dependent. We are developing text-prediction software for the Palm and Pocket PC platforms that uses the language models we have fit to our question corpus. After each new character it presents a fresh list of choices, from which the user can select a whole or partial continuation. Figure 3 depicts a web-based prototype. Once we finalize the interface we intend to measure the throughput and mean time required to input each question.

For now consider the following naïve estimate. Assume we employ the 2-gram language model in Table 3 of 3.4 MB with a perplexity of 59.3 and OOV rate of 3.6%. If the next word to be typed is in the lexicon, the first letter typed reduces the average branching factor from 59.3 by a factor of

Figure 3: A web-based prototype of text prediction for questions based on 3-gram language models. Available at `http://speech.ftw.at/~ejs/answerbus`.

about 26*, to 2–3, and the screen displays the intended word, which is chosen with another tap. The remaining 3.6% of words must be entered in full. So, with a good interface, a rate of 2–3 keystrokes per word might be possible.

5 Implications for entering questions with speech

The first prerequisite for speech input to a mobile device is a programmable microphone. Most Pocket PC devices include these; most Palm devices do not. Networked devices can recognize speech alone or distributedly. Alone, a device must store a language model, phonetic dictionary, acoustic model, and decoder; the decoding algorithm it runs to find the likeliest hypothesis must be efficient enough given the device's memory constraints and speed. In distributed speech recognition the device extracts features from the waveform of the utterance, transmits these to a powerful server using an established protocol like the Aurora DSR protocol of the European Telecommunications Standards Institute [10], and after processing receives a list or lattice of likely hypotheses. There is no reason in principle why a distributed architecture cannot employ speaker-dependent models trained for individual users, although this may in practice be expensive.

In either scenario the interface the device presents should allow quick correction of misrecognized words by keypad or stylus. Schofield and Kubin [1] describe mobile interfaces for posing questions in more depth.

We conjecture that, with a wideband signal and mild background noise, a speech recognizer customized for questions may mistranscribe around 5% of words with speaker-dependent acoustic models, or 10–15% otherwise. The sources [11, 12, 13] report similar word-error rates for tasks of this perplexity or greater.

We hope in the future to build such a recognizer and test various interfaces for efficiently correcting mistranscriptions.

6 Future Work

Language modeling for small devices differs in one essential respect from traditional language modeling: that the space available to store models may be constrained. Various models with fewer parameters than n-grams have been proposed, among which class-based n-grams and maximum-entropy models appear promising for this domain, being intuitively sensible and

*An overestimate. More words beginning e than q allow less disambiguation.

suited to small corpora. We plan to investigate the applicability of these models to natural-language questions in due course.

Accurate language models are necessary but not sufficient for predictive text input. A suitable interface must overcome at least three hurdles: that screens on mobile devices are small; that choosing text from a list requires time, visual attention, and concentration; and that a manual facility for entering uncommon words is necessary. To create an interface supporting easy, rapid entry of text requires careful thought and thorough testing.

For input with speech the decoder must be appropriate to the language models. The time required for Viterbi decoding is proportional to the square of the number of states in the compound hidden Markov model, while the number of states is, for 3-gram language models, itself proportional to the square of the vocabulary size. A tree search, or stack decoding, framework offers more promise for language models of arbitrary complexity. We have investigated approximate tree-search algorithms suitable for speech recognition with complex language models; a paper is forthcoming.

7 Summary

This paper has investigated language models suitable for textual or spoken input of natural-language questions. It has examined a corpus of about 280k unique questions asked by users of the Internet and shown their short-range lexical structure to be more constrained than several corpora like DARPA's Navel Resource Management and the Wall Street Journal. In the light of these results it has discussed the requirements and potential for predictive text input and speech recognition of questions with mobile devices.

8 Acknowledgements

I would like to thank Gernot Kubin and Stefan Rüger for discussions and suggestions. A Marie Curie fellowship from the European Commission supported this research.

References

[1] Edward J. Schofield and Gernot Kubin. On interfaces for mobile information retrieval. In Fabio Paternò, editor, *Proceedings of the 4th Intl. Symposium on Human Computer Interaction with Mobile Devices*

(MobileHCI), number 2411 in Lecture Notes in Computer Science, pages 383–387. Springer-Verlag, September 2002.

[2] K. Hammond, R. Burke, C. Martin, and S. Lytinen. FAQ Finder: a case-based approach to knowledge navigation. In *Proceedings of the Eleventh Conference on Artificial Intelligence for Applications*, pages 80–86, Los Alamitos, February 1995. IEEE Computer Society Press.

[3] Zhiping Zheng. AnswerBus question answering system. In *Human Language Technology Conference (HLT 2002)*, San Diego, CA., March 2002.

[4] Ellen M. Voorhees. Overview of the TREC 2001 question answering track. In Ellen M. Voorhees and Donna K. Harman, editors, *Proceedings of the Tenth Text REtrieval Conference (TREC-10)*. Department of Commerce, National Institute of Standards and Technology, 2001.

[5] Ian H. Witten and Timothy C. Bell. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory*, 37(4), July 1991.

[6] D. Jurafsky and J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice–Hall, 2000.

[7] Philip Clarkson and Tony Robinson. Towards improved language model evaluation measures. In *Proceedings of EUROSPEECH 99, 6th European Conference on Speech Communication and Technology*, volume 5, pages 1927–1930, September 1999. Budapest, Hungary.

[8] George Kingsley Zipf. *The Psycho-biology of Language: An Introduction to Dynamic Philology*. The MIT Press, 1965.

[9] I. Scott MacKenzie. KSPC (keystrokes per character) as a characteristic of text entry techniques. In Fabio Paternò, editor, *Proceedings of the Fourth International Symposium on Human Computer Interaction with Mobile Devices*, number 2411 in Lecture Notes in Computer Science, pages 195–210. Springer-Verlag, September 2002.

[10] D. Pearce. An overview of ETSI standards activities for distributed speech recognition front-ends. In *Proceedings of AVIOS 2000: The Speech Applications Conference*, May 2000.

[11] R. Cole, J. Mariani, H. Uszkoreit, A. Zaenen, and V. Zue. Survey of the state of the art in human language technology. Technical report, Center for Spoken Language Understanding CSLU, Carnegie Mellon University, Pittsburgh, PA, USA, 1996.

[12] F. Zheng and J. Picone. Robust low perplexity voice interfaces. Technical report, MITRE Corporation, August 2001.

[13] Ciprian Chelba. *Exploiting Syntactic Structure for Natural Language Modeling*. PhD thesis, CLSP, The Johns Hopkins University, 2000.