ASSESSING SEARCH TERM STRENGTH IN SPOKEN TERM DETECTION

Amir Hossein Harati Nejad Torbati and Joe Picone

Department of Electrical and Computer Engineering
Temple University, Philadelphia, USA
amir.harati@gmail.com, picone@temple.edu

ABSTRACT

Spoken term detection (STD) is an extension of text-based searching that allows users to type keywords and search audio files containing recordings of spoken language. Performance is dependent on many external factors such as the acoustic channel, the language and the confusability of the search term. Unlike text-based searches, the quality of the search term plays a significant role in the overall perception of the usability of the system. In this paper, we present a system that predicts the strength of a search term from its spelling that is based on an analysis of spoken term detection output from several spoken term detection systems that participated in the NIST 2006 STD evaluation. We show that approximately 57% of the correlation can be explained from the search term, but that a significant amount of the confusability is due to other acoustic modeling issues.

Index Terms— spoken term detection, voice keyword search, information retrieval

1. INTRODUCTION

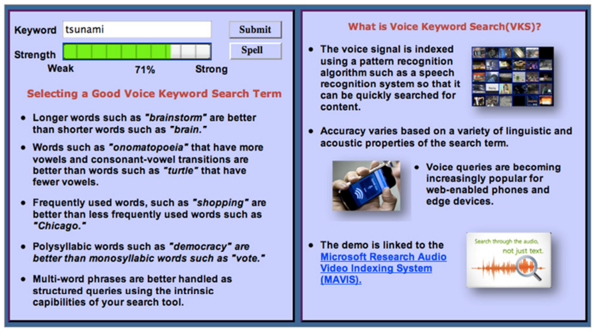
Spoken term detection (STD) systems differ from text search engines in one significant manner – the match between a keyword and the audio data is approximate and is typically based on a likelihood computed from some sort of pattern recognition system. The performance of such systems depends on many external factors such as the acoustic channel, speech rate, accent, language, and the confusability of search terms. In this paper, our focus is on the latter issue. Our goal is to develop algorithms to predict the reliability or strength of a search term using the error rate of the system as measure of the performance.

The accuracy of a search term is a critical issue for frequent users of this technology. Unlike text searches, sorting through audio data that has been incorrectly matched can be a time-consuming and frustrating process. State of the art systems based on this technology produce results that are not always intuitive – a close acoustic match might not necessarily be close in the semantic space. Therefore, the goal of this work is provide users some prior knowledge of which search terms are likely to be more accurate than others. Password strength checkers, which are a similar technology, have become very commonplace, and represent a functional model for this work.

An online demo of the system is available at: *http://www.isip.piconepress.com/projects/ks_prediction/demo/*. A screenshot is shown in Figure 1. Our general approach has been to analyze error patterns produced by existing keyword search systems and to develop a predictive model of these errors. The basis for this work is the NIST Spoken Term Detection (STD) conducted in 2006 [1]. This data is rather unique because we have reference transcriptions for the utterances as well as keyword search results for three of the participants: BBN, IBM and SRI. With such data, we can explore machine learning algorithms that attempt to develop mappings between features derived from the spelling of a keyword and the associated error rate for that keyword. This process is summarized in Figure 2.

Note that this strength prediction function is not just a function of the term's spelling (and other linguistic properties that can be derived solely from the spelling). For example, if a term appears in audio data from a noisy acoustic channel much different than the conditions of the STD evaluation, the error rate associated with that term might not be correctly predicted. Acoustic issues are difficult to represent with this model. One way of overcoming this problem is by marginalizing over all other factors, either by imposing similar conditions for the entire corpus (which is very difficult) or by using a very diverse and large corpus to minimize the average effect of other factors.

In this research, however, we do not have access to the STD systems and we are simply given their output data on various recognition and keyword search tasks. As a result of this limitation, the error rate calculated for each term is not completely marginalized over all other factors and



Selecting a Good Voice Keyword Search Term

- Longer words such as *"brainstorm"* are better than shorter words such as *"brain."*
- Words such as *"onomatopoeia"* that have more vowels and consonant-vowel transitions are better than words such as *"turtle"* that have fewer vowels.
- Frequently used words, such as *"shopping"* are better than less frequently used words such as *"Chicago."*
- Polysyllabic words such as *"democracy" are better than monosyllabic words such as "vote."*
- Multi-word phrases are better handled as structured queries using the intrinsic capibilities of your search tool.

What is Voice Keyword Search(VKS)?

- The voice signal is indexed using a pattern recognition algorithm such as a speech recognition system so that it can be quickly searched for content.
- Accuracy varies based on a variety of linguistic and acoustic properties of the search term.
- Voice queries are becoming increasingly popular for web-enabled phones and edge devices.
- The demo is linked to the Microsoft Research Audio Video Indexing System (MAVIS).

Figure 1. A screenshot of a tool that assesses voice keyword search term strength and displays a confidence measure.

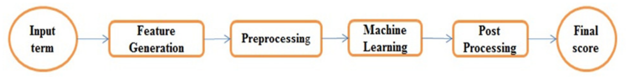


Figure 2. An overview of our approach to search term strength prediction that is based on decomposing terms into features.

effectively contains some noise. Therefore, a secondary goal from this work was to see precisely how much of the performance can be explained simply from automatically generated linguistic information.

2. SEARCH TERM STRENGTH PREDICTION

The goal of a typical STD system [1] is "to rapidly detect the presence of a term in large audio corpus of heterogeneous speech material." STD systems for practical reasons typically index the audio data as a preprocessing step, allowing users to rapidly search the index files using common information retrieval approaches. Indexing can be done using speech to text (STT) systems with phonetic acoustic models [2], or simpler engines based on phoneme recognition [3][2]. The STT approach, which we will focus on in this paper, is summarized in Figure 3.

STD like most detection tasks can be characterized in terms of two kinds of errors: false alarms and missed detections. The first type of error occurs when the system declares an occurrence falsely and the second type occurs when the system does not spot an actual occurrence. In practice, there is always a trade-off between these two kinds of errors and users can tune the system to work according to the application requirements. The overall error could be defined as a linear combination of these two terms. In this paper, we give equal weights to both factors.

A summary of our approach is given in Figure 2. The preprocessing block performs normalization. Post-processing is an optional block to convert the predicted error to a more proper score (i.e. score between 0-100) using a deterministic function. A key element of this work is the machine learning algorithm. We investigated three approaches using standard MATLAB toolboxes [4]: linear regression (*regress*), a feed-forward neural network (*Neural Network Toolbox*) and regression trees (*classregtree*).

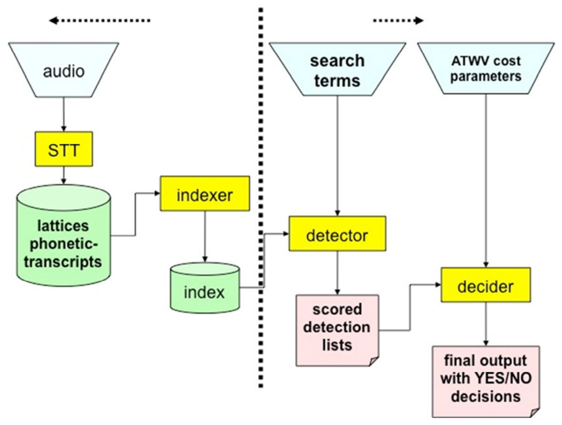


Figure 3. An overview of a common approach to voice keyword search that uses an STT system to perform indexing [2].

A fourth approach based on a *k*-nearest neighbor (KNN) regression was also developed. An intuitive approach to estimate the error associated to a new term is to look at structurally similar terms that exist in the training data and average their associated errors. This is known as *k*-nearest neighbor regression. The only difficulty is to find structurally similar terms. We use dynamic programming to find the distance between different terms that can be calculated based on phonetic transcriptions. In this way, we can find phonetically similar terms in the training dataset and use them to estimate the error for the new term.

The first step in the process, feature generation, is perhaps the most interesting portion of this work. In order to use the above algorithms, features should be extracted from the spellings of the keywords. The first step is to convert terms into phonetic representations using a combination of dictionaries and letter-to-sound rules [5]. Converting to broad phonetic class (BPC) and consonant-vowel-consonant (CVC) representations can be done easily using a conversion table. Other features such as duration, type of initial and final phonemes, number of vowels and consonants, ratio of vowels to consonants, number of letters, number of syllables and also different N-gram representations can be calculated using the basic phonetic representation.

Some features are more subtle to compute. An initial hypothesis in this work is that duration plays an important role in search term accuracy[1]. Duration can be estimated by using a reliable dataset of phonetically-segmented data to construct a table for the average duration of all N-grams. By converting each term into its N-gram representation we can simply compute its average duration using this look-up table. Here we have used a simpler approach based on the monophone representation. Syllable counts, which are correlated with duration, were computed using dictionary lookup and syllabification software [6].

Each search term was converted to its feature representation using the above approaches. These features represent the input to the process. The search term error rate was extracted from the NIST 2006 evaluation results. This represents the desired output. Our experiments focused on predicting error rates from these features.

3. EXPERIMENTATION

Our experiments were implemented using MATLAB. Both closed loop and open loop tests were conducted. For the closed loop case all data was used for both training and testing. For the open loop case, data has been divided into two disjoint sets (80% for training 20% for testing). This partitioning was created randomly and repeated 100 times. The results of these 100 independent experiments were averaged to produce our error rate estimates. Mean square error (MSE) and correlation (R) were used to measure the performance of the algorithms.

Four datasets were used to generate the experimental data used for the prediction function. First, we began with the NIST 2006 evaluation results. This dataset consists of evaluation results for three different sites [1]: IBM, BBN and SRI and for three different kinds of sources: broadcast news, telephone conversations and conference meetings. To reduce the effect of noise we have selected terms that occurred at least three times in the dataset.

This data was supplemented by speech recognition output on the Fisher corpus provided by BBN. This data set was significantly larger than the NIST dataset, and we limited it to terms that occurred at least ten times.

Third, we used speech recognition output from the WSJ task generated by the Aurora baseline system [7]. We have used terms that occurred at least three times in the dataset. Finally, to estimate duration, we used the TIMIT phonetically transcribed dataset [8].

The first correlate we explored was duration, which our practical experience tells us is significant. In Figure 4, we explore error rate as a function of word duration, and confirm that as a first-order effect, duration is important. Several features, however, demonstrate this type of relationship. In Table 1 through Table 3, we show results for selection of feature sets on three different corpora.

In these tables, rows correspond to feature combinations. For example, the third row in Table 1 is labeled "Duration+Syll" which indicates that this feature set combines duration and the number of syllables. "Cons" refers to the number of consonants and "Vowels" refers to the number of vowels. A compound term like "BPC Bigrams" refers to bigrams of broad phonetic classes (BPC). "CVC" refers to a reduction of each phoneme label to its consonant or vowel class.

From these tables we can conclude that fairly simple feature combinations resulted in a correlation close to 0.5. Performance does not improve significantly when more feature combinations are added, indicating there is little new information in these features.

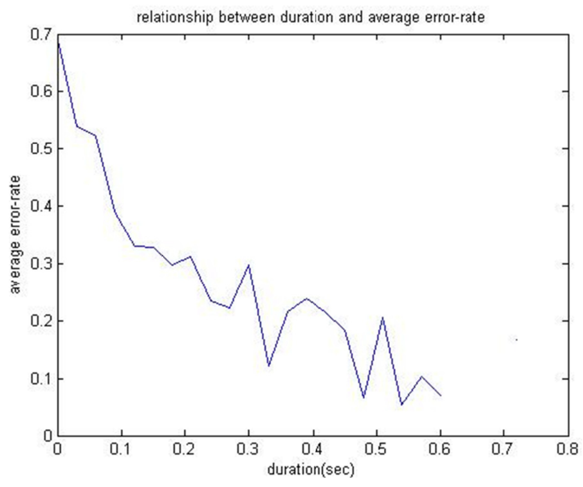


Figure 4. The relationship between duration and error rate shows that longer words generally result in better performance.

We see from the first three tables that performance on the NIST and BBN data is different from WSJ. This is an indication of intrinsic differences between STD and speech recognition systems. Best performance obtained for WSJ with R=0.57. The best performance for BBN gives R=0.53. This most likely relates to the more carefully controlled speech rate for the WSJ corpus. Generally, we can see the BBN data set gives better results and is stable over a variety of algorithms and feature sets. The reason for this

Table 1. Performance of a variety of feature combinations on the NIST 2006 data for three different learning algorithms.

Features	Data Source: NIST 2006											
	Closed-Loop						Open-Loop					
	LR		NN		RT		LR		NN		RT	
	MSE	R	MSE	R	MSE	R	MSE	R	MSE	R	MSE	R
Duration	0.04	0.46	0.06	0.43	0.04	0.48	0.05	0.46	0.06	0.41	0.05	0.45
#Syll	0.05	0.28	0.07	0.23	0.05	0.28	0.05	0.28	0.07	0.22	0.05	0.27
Duration+#Syll	0.04	0.46	0.05	0.45	0.04	0.53	0.05	0.46	0.06	0.38	0.05	0.46
#Syll+#Cons	0.05	0.32	0.07	0.29	0.05	0.41	0.05	0.31	0.07	0.24	0.05	0.31
Duration+#Syll+#Cons	0.04	0.46	0.06	0.43	0.04	0.60	0.05	0.46	0.06	0.37	0.05	0.41
Duration+Length	0.04	0.46	0.06	0.44	0.04	0.55	0.05	0.46	0.07	0.38	0.05	0.41
Duration+Length/Duration	0.04	0.47	0.06	0.46	0.02	0.76	0.05	0.47	0.06	0.44	0.07	0.28
Duration+#Syll+Length/Duration	0.04	0.47	0.05	0.47	0.02	0.79	0.05	0.46	0.05	0.42	0.06	0.33
Duration+#Cons+Length/Duration	0.04	0.47	0.05	0.47	0.02	0.78	0.05	0.46	0.05	0.42	0.07	0.31
Duration+#Vowels/#Cons+Length/Duration	0.04	0.47	0.05	0.47	0.02	0.80	0.05	0.46	0.05	0.43	0.07	0.31
Duration+#Cons+#Vowels/#Cons	0.04	0.46	0.06	0.44	0.04	0.58	0.05	0.45	0.06	0.37	0.06	0.32
BPC BIGRAMS(36)	0.05	0.38	0.06	0.29	0.02	0.77	0.06	0.23	0.08	0.08	0.08	0.12
CVC BIGRAMS +TRIGRMAS	0.05	0.35	0.06	0.29	0.04	0.50	0.05	0.29	0.07	0.15	0.06	0.19
CVC MONOGRAMS +BIGRAMS+TRIGRAMS	0.05	0.35	0.07	0.28	0.04	0.50	0.05	0.29	0.08	0.15	0.06	0.18
Duration+#Cons+Length/Duration+CVC1	0.04	0.47	0.05	0.45	0.02	0.79	0.05	0.46	0.06	0.41	0.07	0.31
Duration+#Cons+Length/Duration+CVC4	0.04	0.47	0.05	0.45	0.02	0.78	0.05	0.45	0.06	0.38	0.07	0.30
Duration+#Cons+#Syll/Duration+CVC3	0.04	0.47	0.05	0.46	0.02	0.79	0.05	0.46	0.06	0.41	0.07	0.29

Table 2. Performance of the same features is shown for the BBN Fisher data. Results correlate well with the NIST STD data.

Features	Data Source: BBN											
	Closed-Loop						Open-Loop					
	LR		NN		RT		LR		NN		RT	
	MSE	R	MSE	R	MSE	R	MSE	R	MSE	R	MSE	R
Duration	0.03	0.40	0.03	0.44	0.03	0.42	0.03	0.40	0.03	0.43	0.03	0.42
#Syll	0.03	0.47	0.05	0.42	0.03	0.49	0.03	0.47	0.05	0.42	0.03	0.49
Duration+#Syll	0.03	0.48	0.04	0.48	0.03	0.52	0.03	0.48	0.04	0.47	0.03	0.51
#Syll+#Cons	0.03	0.49	0.05	0.46	0.03	0.52	0.03	0.49	0.05	0.45	0.03	0.51
Duration+#Syll+#Cons	0.03	0.49	0.03	0.51	0.03	0.55	0.03	0.49	0.03	0.50	0.03	0.50
Duration+Length	0.03	0.48	0.03	0.52	0.03	0.53	0.03	0.48	0.03	0.51	0.03	0.50
Duration+Length/Duration	0.03	0.49	0.04	0.49	0.03	0.61	0.03	0.49	0.04	0.48	0.04	0.42
Duration+#Syll+Length/Duration	0.03	0.50	0.03	0.53	0.02	0.66	0.03	0.50	0.03	0.52	0.04	0.42
Duration+#Cons+Length/Duration	0.03	0.50	0.04	0.53	0.02	0.68	0.03	0.50	0.04	0.52	0.04	0.41
Duration+#Vowels/#Cons+Length/Duration	0.03	0.50	0.03	0.54	0.02	0.67	0.03	0.50	0.03	0.53	0.04	0.41
Duration+#Cons+#Vowels/#Cons	0.03	0.49	0.04	0.51	0.03	0.56	0.03	0.49	0.04	0.50	0.03	0.51
BPC_BIGRAMS(36)	0.03	0.51	0.03	0.53	0.02	0.70	0.03	0.50	0.03	0.48	0.04	0.41
CVC BIGRAMS +TRIGRMAS	0.03	0.53	0.04	0.50	0.03	0.56	0.03	0.52	0.04	0.48	0.03	0.51
CVC MONOGRAMS +BIGRAMS+TRIGRAMS	0.03	0.53	0.03	0.54	0.03	0.57	0.03	0.52	0.03	0.52	0.03	0.51
Duration+#Cons+Length/Duration+CVC1	0.03	0.50	0.03	0.55	0.02	0.70	0.03	0.50	0.03	0.53	0.04	0.40
Duration+#Cons+Length/Duration+CVC4	0.03	0.50	0.03	0.53	0.02	0.69	0.03	0.50	0.03	0.52	0.04	0.40
Duration+#Cons+#Syll/Duration+CVC3	0.03	0.49	0.04	0.50	0.02	0.66	0.03	0.49	0.04	0.48	0.04	0.43

Table 3. Performance on the WSJ dataset.

Features	Data Source: WSJ											
	Closed-Loop						Open-Loop					
	LR		NN		RT		LR		NN		RT	
	MSE	R	MSE	R	MSE	R	MSE	R	MSE	R	MSE	R
Duration	0.02	0.06	0.02	0.40	0.01	0.66	0.02	0.06	0.02	0.35	0.01	0.56
#Syll	0.02	0.25	0.02	0.18	0.02	0.26	0.02	0.25	0.02	0.18	0.02	0.26
Duration+#Syll	0.02	0.28	0.02	0.40	0.01	0.70	0.02	0.28	0.02	0.35	0.01	0.55
#Syll+#Cons	0.02	0.27	0.03	0.22	0.02	0.35	0.02	0.27	0.03	0.18	0.02	0.25
Duration+#Syll+#Cons	0.02	0.31	0.02	0.38	0.01	0.77	0.02	0.30	0.02	0.27	0.02	0.46
Duration+Length	0.02	0.31	0.02	0.45	0.01	0.73	0.02	0.32	0.02	0.37	0.01	0.50
Duration+Length/Duration	0.02	0.28	0.02	0.49	0.01	0.70	0.02	0.29	0.02	0.41	0.01	0.57
Duration+#Syll+Length/Duration	0.02	0.29	0.02	0.28	0.01	0.72	0.02	0.28	0.02	0.21	0.01	0.52
Duration+#Cons+Length/Duration	0.02	0.30	0.02	0.41	0.01	0.74	0.02	0.29	0.02	0.30	0.01	0.52
Duration+#Vowels/#Cons+Length/Duration	0.02	0.29	0.02	0.40	0.01	0.75	0.02	0.28	0.02	0.32	0.01	0.52
Duration+ #Cons+ #Vowels/#Cons	0.02	0.31	0.02	0.44	0.01	0.77	0.02	0.31	0.02	0.37	0.02	0.48
BPC_BIGRAMS(36)	0.02	0.40	0.02	0.22	0.01	0.66	0.02	0.21	0.03	0.05	0.03	0.12
CVC BIGRAMS +TRIGRMAS	0.02	0.40	0.02	0.27	0.02	0.45	0.02	0.32	0.02	0.16	0.02	0.23
CVC MONOGRAMS +BIGRAMS+TRIGRAMS	0.02	0.40	0.02	0.25	0.02	0.45	0.02	0.32	0.02	0.15	0.02	0.23
Duration+#Cons+Length/Duration+CVC1	0.02	0.31	0.02	0.38	0.01	0.78	0.02	0.30	0.02	0.27	0.02	0.48
Duration+#Cons+Length/Duration+CVC4	0.02	0.30	0.02	0.38	0.01	0.75	0.02	0.28	0.02	0.28	0.02	0.49
Duration+#Cons+#Syll/Duration +CVC3	0.02	0.34	0.02	0.39	0.01	0.76	0.02	0.33	0.02	0.31	0.01	0.53

phenomenon is related to the size of the BBN data set, allowing noise reduction through averaging.

It can be seen that "duration" is the single most important predictor for all data and algorithms. Many other features, like length and number of syllables, are effectively different approximations of duration. This is the reason that adding these features together does not improve performance significantly. However, the latter statement is not true for features like BPC and CVC N-grams. As it can be seen from these tables, these features also give comparable results to the best result accessible by duration and its approximations. The last three rows of these tables show the results of using duration and its related features along with some CVC monograms.

Table 4 shows the result for the KNN-based algorithm for the BBN data. Due to KNN's need for large data, only the data from the BBN set was used. For each new data point, we compute its distance from all other points. This computation is achieved using a dynamic programming algorithm in which we try to find the smallest edit distance [9][8] between the given phonetic representation of a new term and all phonetic representations of terms existing in training data. K is the number of points that we use to compute error for the new term by averaging. In this algorithm we use a weighted averaging method in which the weight of each data point is reciprocal to its distance from the new data point. The results for $K > 3$ are comparable to the first three algorithms. Table 4 shows K controls the error rate for both open loop and closed loop cases and there is a tradeoff between these error rates.

Table 4. Performance on the BBN data using the KNN algorithm. As expected, the closed-loop results are good, and open loop performance increases with the number of clusters.

K	Data Source: BBN			
	Closed-Loop		Open-Loop	
	MSE	R	MSE	R
1	0.00	0.97	0.05	0.32
3	0.02	0.74	0.03	0.43
100	0.03	0.54	0.03	0.53
400	0.03	0.53	0.03	0.51

4. CONCLUSION

In this paper, the problem of assessing search term strength for STD systems has been introduced. Four relatively simple algorithms were explored using a wide variety of feature combinations. The best open loop performance achieved a maximum R value of approximately 0.5. This leads us to believe there are limits to the ability of the proposed prediction paradigm to account for recognition errors. Further refinement of the predictor can come from either an improved machine learning algorithm or incorporation of more knowledge of the acoustic models used in the recognition engines. We are currently exploring hierarchical Bayesian frameworks to construct a more powerful predictor.

5. ACKNOWLEDGEMENTS

We gratefully acknowledge three 2006 STD sites for providing the data referenced in this paper: Owen Kimball (BBN), Bhuvana Ramabhadran (IBM) and Dimitra Vergyi (SRI). We are also indebted to Jon Fiscus (NIST) and colleagues at DoD for enabling this research by providing the corresponding 2006 STD data and associated support.

6. REFERENCES

[1] J. G. Fiscus, *et al.*, "Results of the 2006 spoken term detection evaluation," *Proc. Workshop Searching Spont. Conv. Speech*, pp. 45–50, Amsterdam, NL, July 2007.

[2] D. Miller, *et al.*, "Rapid and Accurate Spoken Term Detection," *Proceedings of INTERSPEECH*, pp. 314-317, Antwerp, Belgium, Sep. 2007.

[3] *http://www.nexidia.com/technology/phonetic_search_technology*

[4] *http://www.mathworks.com*

[5] H.S. Elovitz, *et al.*, "Automatic Translation of English Text to Phonetics by Means of Letter-to-Sound Rules", NRL Report 7948, Naval Research Laboratory, Washington, DC, 1976.

[6] W. M. Fisher, "tsyl: NIST Syllabification Software," available at *http://www.nist.gov/speech/tools*, June 1997.

[7] N. Parihar, *et al.*, "Performance Analysis of the Aurora Large Vocabulary Baseline System," *Proc. of the 12th European Signal Proc. Conf.*, pp. 553-556, Vienna, Austria, Sept. 2004.

[8] W. Fisher, *et al.*, "The DARPA Speech Recognition Research Database: Specifications and Status," *Proceedings of DARPA Workshop on Speech Recognition*. pp. 93–99, February 1986.

[9] R. Wagner and M. Fischer, "The String-to-String correction problem," *J. ACM*, vol. 21, no. 1, pp. 168–173, Jan. 1974.