Key Word Spotting for Low Resource Languages

Student ID: J902G
Word Count: 2,975[a]
Submitted: March 23, 2022

[a]Excluding footnotes, algorithms, and references.

Contents

1 Introduction

This report investigates methods for improving key word spotting in the low resource language of Swahili, where Automatic Speech Recognition (ASR) systems perform poorly due to a shortage of labelled training data. The metric used throughout to gauge Key-Word Spotting (KWS) system performance is the Term Weighted Value (TWV), defined as:

$$\text{TWV}(\theta) = 1 - P_M(\theta) - \beta P_{FA}(\theta) \tag{1}$$

where β is set to 999.9 for this practical, and in the IARPA Babel program. From here, the actual term-weighted value (ATWV) is defined as the TWV for the chosen system's detection threshold, $\hat{\theta}$, such that ATWV = TWV($\hat{\theta}$) [5]. The Maximum TWV (MTWV) is then defined as the highest TWV over all choices of θ. The range of TWV values is thus $-\infty$ to 1, where 1 is the value when no there are no false alarms or misses, and -∞ when the number of false alarms tends to infinity [5]. For the purpose of this report, we will focus on MTWV scores, as this means that we do not need to explicitly define the threshold to use for each case, and makes different systems easier to compare. The difference between MTWV and ATWV values is still useful, however, as a gauge of how many hits were found by the KWS system (if there are very few hits found then the setting of θ matters less in deciding between hits of a single key word, and therefore the ATWV and MTWV values are more similar). This can also be seen visually from the legnth of the tail on the DET curves - as we do not set a threshold value, the ATWV is calculated using the lowest possible miss probability, which for long tails (caused by many hits per key word) corresponds to a high probability of false alarm and a lower ATWV.

2 Indexing

The first step in the KWS system is to build a means of deciding whether a query in the query file is a hit with the ASR output or not. The means of deciding whether a phrase is a hit consists of the following conditions:

1. All of the words in the phrase are included in the same file with the same channel, and in the correct time-order
2. The time difference between words in the query is less than 0.5 seconds

The indexer algorithm to index each query is therefore:

```
initialise index = {}
find K hits of the first query word
phrase = first query word
duration = duration of first query word
score = score of first query word
beginning time = beginning time of first query word

for word i in the query:
    for k <= K:
        if word (i + 1) is after word i in the ASR output for filename of hit k:
            if time gap between word i and word (i + 1) < 0.5:

                append word (i + 1) to phrase
                duration = duration + duration (i + 1)
                score = score * score (i + 1)

                append [filename, channel, duration, beginning time, phrase, score] to
                    index[kwid]

            end if
        end if
    end for
end for
```

Listing 1: Algorithm for making the decision on whether a query phrase is a hit of the ASR output, and putting it in the index if it is.

Therefore, in terms of algorithmic cost, if the entire query is found in the ASR output, the first *for* loop will iterate N times for N words in the query and the second will iterate K times for K hits of the first query word, meaning that the algorithmic complexity is $\mathcal{O}(NK)$ per query in the queries file.

3 Morphological Decomposition

Using the indexer as described above on the supplied 1-best ASR output[1] gave the decode.ctm MTWV results shown in the grey shaded row of Table 1. The IV MTWV is lower than the optimal

[1] 1-best outputs refer to ASR outputs that only include the highest scoring prediction from the spoken utterance, rather than an n-best list of outputs.

ground truth for that query file (TWV = 1) due to detection errors in the ASR output stored in decode.ctm[2]. The total MTWV is also brought down by the fact that the OOV MTWV score is 0, i.e. the out of vocabulary words have a 100% probability of missing and the probability of false alarm is 0, such that TWV = 0 from Equation (1). This is because if a query phrase is OOV due to an ASR misdetection, then a simple indexing method as thus far described will never decide a hit for that query. In order to improve OOV rates, we therefore need a means of altering the query and ASR output so that the OOV phrases can be adjusted in a way that allows them to form a hit despite ASR errors. One means to do this is via morphological decomposition.

Two types of morphological decomposition are included in Table 1 below. The first uses the previous decode.ctm ASR system output and the original query file and morphologically maps both using morph.dct and morph.kwslist.dct respectively. The purpose of this is to pick up errors of the ASR system where the correct root of the word is used but a hit not detected due to the ASR system giving it the wrong grammatical structure (for example, an *'-ed'* added where it should not be). The second version uses the same morphed query file but an automatically morphed ASR output as the ASR system has access to the morphologically decomposed vocabulary at the utterance classification stage. This means that the ASR output does not need additional morphological mapping.

System	All	IV	OOV
decode	0.3191	0.4021	0.000
decode + morph	0.3106	0.387	0.018
decode-morph	0.3172	0.382	0.066

Table 1: The grey shaded row uses the original non-morphed ASR output with the non-morphed queries. Unshaded rows then show the MTWV values for IV, OOV and IV+OOV phrases in the two types of morphological decomposition.

The results demonstrate that morphological decomposition does indeed improve OOV MTWV values, indicating that there exist some phrases that are not originally a hit, but that contain one or more sub-phrases that are a hit. The morphological decomposition however decreased IV MTWV, as the inclusion of morphs caused there to be more hits which increases the false alarm probability. The decode-morph.ctm ASR output gives higher OOV and overall performance than manual morphing of the decode.ctm output indicating that performing manual morphological decomposition is not as effective as using a morph-based system. This is as expected, as the output of the non-morph-based ASR is more likely to be incorrect in the first place if the ASR does not have access to the morph vocabulary.

It was important for morphological decomposition to consider how duration and score are handled when a phrase is split apart. The duration is simple to handle, and the original phrase duration simply divided uniformly by the number of sub-phrases in the original phrase (this is essentially assuming that each sub-phrase has the same duration). The score calculation is more involved and is inspired by working backwards on how scores would be calculated for a net phrase if sub-phrases were to be combined. Assuming each sub-phrase has equal score, when multiple hits are combined their scores are multiplied to give $s_1 \cdot s_2 \cdot s_3 ... = s^{\#(\text{sub-phrases})}$. Therefore, doing the reverse means that if s_T is the score of a phrase, each sub-phrase score is:

$$s = s_T^{1/\#(\text{sub-phones})} \tag{2}$$

[2]The fact that the language being used for the ASR system is low-resource means that the ASR accuracy will be low due to a lack of training data.

4 Score Normalisation

In its original form, the score associated to a key word hit is an estimate for the posterior probability of being a hit, given by the ASR system [3]. The original scores are therefore lower for rarer words, even if they have been correctly classified, because the ASR system has seen less of them and is therefore less confident on them being correct. We therefore wish to have a way to increase the scores of rarer words to make them more likely to be above the decision threshold. In addition, by the definition of the TWV, the improvement in TWV is inversely proportional to the frequency of a word in the ASR system and the reduction in TWV due to false alarms independent of the word frequency [4], so it is again conducive to set the scores of the key words so that rarer words are prioritised compared to more frequent ones[3] [3]. An intuitive way to do this is therefore to down-scale scores of words that occur a lot in the ASR output and up-scale words that occur less often via sum-to-one normalisation [6]:

$$\hat{s}_{ik}(\gamma) = \frac{s_{ik}^{\gamma}}{\sum_j s_{ij}^{\gamma}} \tag{3}$$

where s_{ik} is the posterior probability for the k^{th} occurrence of key word i. The setting of γ in Equation (3) controls the polarisation of the scores. For each individual key word, if $\gamma \to 0$ all of that key word's occurrences tend to the same score, and are all therefore small and less likely to be above a set threshold if there is a high occurrence for that word. In contrast, if γ is large, then for each key word the highest scored occurrence is greatly amplified relative to the other occurrences, and thus the KWS system hones in on occurrences that the ASR system was most confident about, with the extent of this polarisation being greater for higher-frequency words.

Figure 1 shows plots of the MTWV as a function of γ from 0 to 4 for IV (left) and IV + OOV (right) phrases in decode.ctm (OOV is omitted as it is 0 for all γ due to no morphological decomposition). The best performance is found to be $\gamma = 3$.

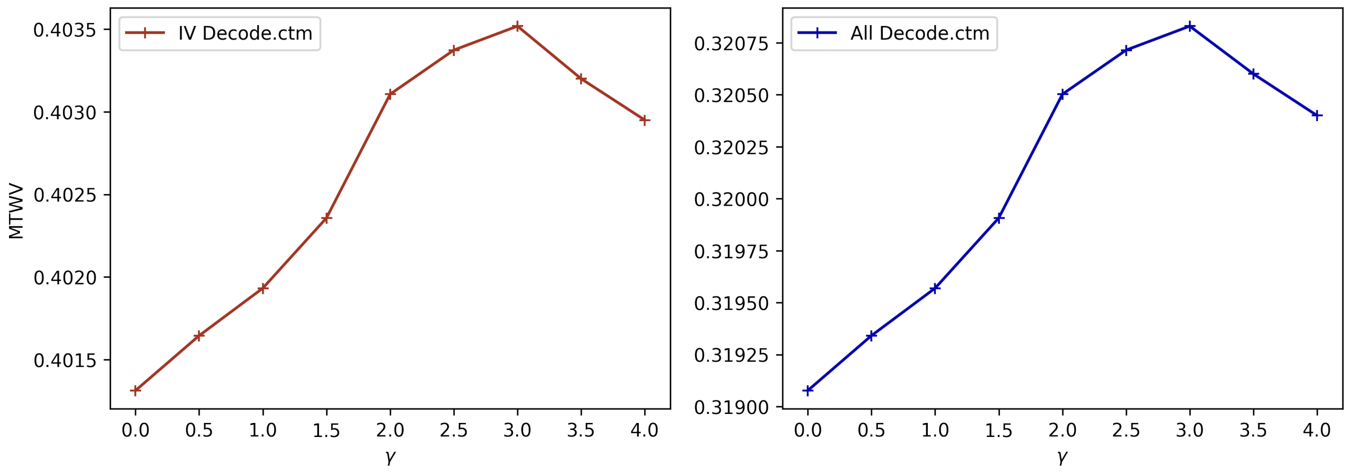

Figure 1: Plot of the MTWV values with the STO exponenent, γ, for IV, OOV, and IV+OOV phrases for the decode.ctm 1-best ASR output. No morphing has been used in these figures.

When the same process is repeated for the morphological decomposition KWS system a different relationship is observed however. This can be seen in Figure 2, where the optimal γ value is now $\gamma = 1$, indicating that the morphed system works better when STO normalisation is less polarising and keeps scores of occurrences closer in value. As discussed before, now that morphological decomposition is

[3]If a word has very high frequency then the reward/cost payoff between hits and false alarms is a lot worse than if the word has low frequency.

used, the OOV MTWV is non-zero, but does not change with γ, indicating that only a single occurrence of any one decomposed OOV sub-phrase is found (which means that STO always sets that occurrence's score to 1 regardless of γ value). With $\gamma = 1$, the highest MTWV so far in the report is found of 0.3240 showing an increase from the previous-best non-morphed STO normalisation at $\gamma = 3$.

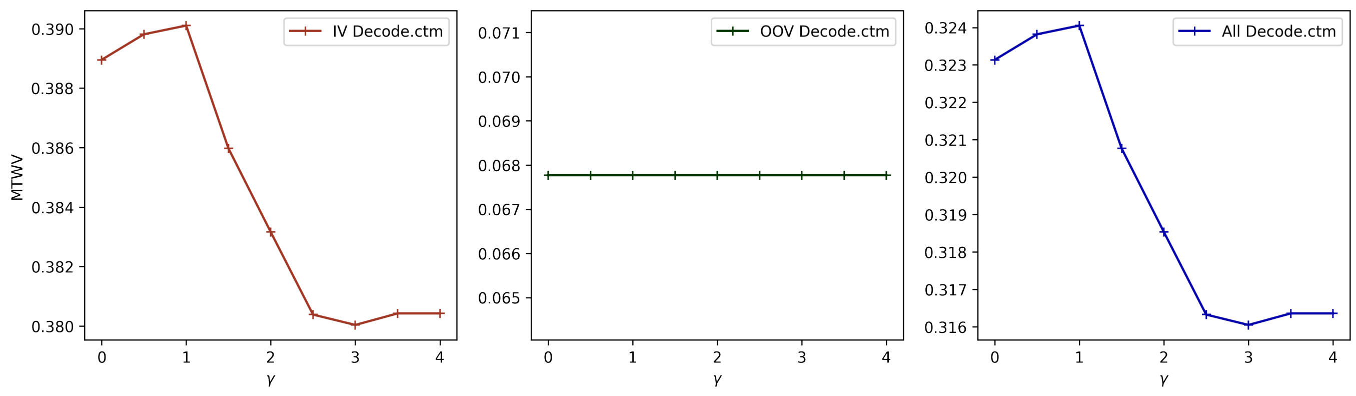

Figure 2: Similar plot of MTWV values with γ as in Figure 1. The decode.ctm file was now morphologically decomposed prior to score normalisation.

Other forms of score normalisation are used for different desired properties; for example, the approach of [1] uses the closeness in time of phones within an occurrence to gauge the score, and prioritises occurrences with longer duration. A similar view is presented in [3] and referred to as Query-length Normalisation. Both of these methods favour duration as a proxy for likelihood of an occurrence being correct over key word frequency, and thus do not seek to explicitly optimise the TWV in the same way that Equation (3) does. Therefore, for this report, STO normalisation is taken as the preferred method.

5 Grapheme Confusion Matrices

In order to improve OOV performance a different strategy to morphological decomposition can also be taken. Grapheme confusion matrices map an OOV string to an IV one by scoring changes from each reference character in it to a hypothesis character from an IV string, in order to find the IV string that has the minimum edit distance. This is also referred to in the literature as the Levenshtein distance between two strings, and is calculated according to the weighted minimum edit distance algorithm below [7].

```
distance[0][0] = 0

for char i in OOV term:
    deletion_cost = 1 - C(i, silence) # 1 - P(hypothesis | silence)
    distance[i][0] = distance[i-1][0] + deletion_cost
end for

for char j in IV term:
    insertion_cost = 1 - C(silence, j) # 1 - P(silence | reference)
    distance[0][j] = distance[0][j-1] + insertion_cost
end for
```

```
for char i (i > 1) in OOV term:
    for char j (j > 1) in IV term:

        substitution_cost = 1 - C(char i, char j) # 1 - P(hypothesis | reference)
        insertion_cost = 1 - C(silence, char j) # 1 - P(silence | reference)
        deletion_cost = 1 - C(char i, silence) # 1 - P(hypothesis | silence)
        distance[i][j] = min(distance[i-1][j] + deletion_cost,
                             distance[i][j-1] + insertion_cost,
                             distance[i-1][j-1] + substitution_cost)

    end for
end for
Edit_distance = distance[-1][-1]
```

Listing 2: Algorithm to calculate the minimum edit distance between an OOV and IV string, using the substitution probability information provided in grapheme.map to calculate insertion, deletion, and substitution costs.

In the algorithm above, the probability of inserting a hypothesis character given the current reference was calculated as the number of times that reference was mistaken by the hypothesis divided by all occurrences of that reference, with the counts given in the grapheme.map file. The mappings made are therefore specific to the language being considered (Swahili in this case), and mean that the change from OOV to nearest IV words make intuitive sense. For example, *nipelekwe* is mapped to *nipeleke* in the non-morphed decode.ctm file, and *ra el ra el* to *ra ol ra ol* in the decode-morph.ctm ASR output. In contrast, the mapping of English phrases make less sense as the grapheme.map file data is not tuned to English phrases, and so it is harder to find a close IV term. For example, *what she has gone* is mapped to *that the hah uone*, which is a lot further from the original than the other cases.

Table 2 below shows the results for the MTWV after grapheme mapping. It shows that grapheme mapping gives higher MTWV values than morphological decomposition alone without STO normalisation, including higher OOV MTWV. This suggests that probabilistic treatment of the phrases in the query performs slightly better, although the difference is small. When STO normalisation is included however, the performance of grapheme mapping is very similar to that of morphological decomposition, as seen in Figure 2 (Right). A shortcoming of grapheme mapping is that it increases the probability of false alarms, especially in the case where an OOV term is mapped to an IV one that is a large edit distance away (for example, in the cases of the English phrases). Therefore, if there was a means to only map OOV terms to IV ones if the edit distance was smaller than some threshold it may present an improvement to the TWV, due to false alarm rates being decreased whilst retaining the benefits of mapping the OOV terms with low edit distances to IV ones.

One main advantage of grapheme mapping is how it is possible to incorporate lattice-based methods in a way not possible using morphological decomposition. This is because grapheme mapping maps each OOV string to a list of IV suggestions ranked by minimum edit distance, and so the process is readily extendable to include N-best IV phrases rather than the 1-best output used here. If lattice methods were to be used here then the need to settle on a single decision for the closest IV term can be delayed, allowing for greater acoustic variation to be accounted for by the lattice terms, and making the KWS system less prone to mapping OOV terms to false alarm IV ones [2].

One disadvantage of grapheme mapping compared to morphological decomposition however is the algorithmic cost, with the algorithm shown above having complexity $O(nmk)$ for n characters in the OOV string, m in the IV string, and k OOV strings to process. In comparison, morphological mapping only has to loop a single time through the query file to map each query phrase to its decomposed variant, and is therefore algorithmically cheaper when there are many OOV terms.

System	All	IV	OOV	Threshold
decode + STO ($\gamma = 3$)	0.3209	0.4036	0	0.007
decode-morph	0.3232	0.3890	0.0678	0.061
decode-morph + STO ($\gamma = 1$)	**0.3243**	**0.3904**	**0.0678**	**0.061**
decode + morph	0.3153	0.3917	0 .0180	0.043
decode + morph + STO ($\gamma = 1$)	0.3154	0.3920	0 .0180	0.017

Table 2: Table of MTWV values when using grapheme confusion matrices to map OOV words in the query to IV words prior to passing the query into the KWS system. For the manual morphological decomposition + grapheme confusion matrix system, the set of IV words is taken to be the set of morphs in the index *after* morphological decomposition on the ASR output files.

6 System Combination

In low resource languages, the lack of training data and consequent poor performance of ASR systems means that many hits are missed due to incorrect classification of speech utterances. However, where one system may miss-classify a spoken utterance, another may classify it correctly, and so using the combined outputs of an ensemble of systems can improve performance. In order to do this combination, KWS system outputs were combined by merging their hit data if the hit phrase is the same for both, and the beginning times of each hit overlap. The scores are then combined by summing the individual scores. If either system contains a hit that the other does not then the hit is carried through to the combined output, but without changing the score (so its score will likely be lower than scores of hits that were in more than one system).

Figure 3 shows the MTWV performance for combinations of systems that have been considered thus far in the report, and compares the combined MTWV to the highest MTWV over the individual systems. The system combination IDs in Figure 3 refer to the following:

1. decode + (decode + morph)
2. decode + (decode + morph) + decode-morph
3. (decode + STO with $\gamma = 3$) + (decode + morph + STO with $\gamma = 1$) + (decode-morph + STO with $\gamma = 1$)
4. (decode + STO with $\gamma = 3$) + (decode-morph + STO with $\gamma = 1$)
5. (decode + STO with $\gamma = 3$) + (decode-morph + STO with $\gamma = 1$) + (decode + grapheme)

From the figure it is apparent that combining systems makes a large positive difference for some systems compared to the non-combined system MTWVs, but negatively affects others. For example, combining the word-based and manual morph-based (decode + morph) systems degrades performance from the word-based system alone before decode-morph is also included (ID 1 to ID 2). STO normalisation also improves the MTWV values (ID 2 to ID 3). In addition, it seems that it is possible to plateau the MTWV when combining many systems, as can be seen moving from ID 4 to ID 5 in the figure, where the only difference in the systems is that graphemes are additionally added in ID 5. The reasoning behind this could be that although more hits are detected when the systems are combined, the false alarms from each system are also carried through, and so the net effect is that there is not much change in the overall MTWV.

In the combination, the scores from each system are added with the same weighting, which assumes that each system in the combination is as good as the others. If this assumption does not seem to

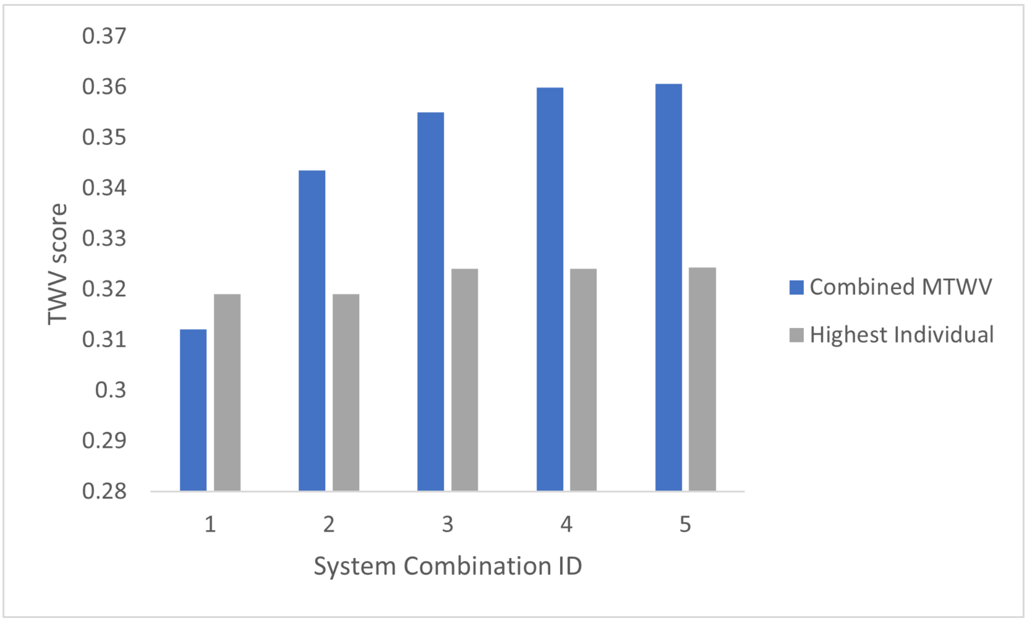

Figure 3: Plot of combined and highest individual MTWV scores for combinations of systems considered up to now in the report. The system ID references can be seen above.

hold then a smarter combination could be considered by weighting each system according to their MTWV values as proposed in [3]. This assigns a weight ω_i to system i in the sum, calculated via:

$$\omega_i = \frac{\text{MTWV}_i}{\sum_j \text{MTWV}_j} \tag{4}$$

Whilst this method may be able to slightly reduce the number of false alarms by better ignoring individual systems that have high false alarms and consequent low MTWV, it would likely not make much difference in this situation, as the MTWVs of each individual system are already very similar (all around 0.3200), and so the weighting would approximately be equal once more. However, if we were considering individual systems with very different performance (for example combining these systems with the IBM lattice KWS systems discussed in Section 6.1 below) then this would be a sensible approach to take.

6.1 IBM WFST Systems

IBM's Weighted Finite State Transducer (WFST) KWS systems were also provided in order to compare their individual and combined performances to the 1-best ASR systems considered so far. Table 3 shows the MTWV values before score normalisation, and the higher MTWVs demonstrate the advantage of lattice systems compared to 1-best output systems. The highest MTWV individual system was word-sys2 and the lowest morph, due to the very high IV MTWV achieved by the word lattice systems compared to the morph-based one. However, as found previously in the report, morph-based systems have better OOV performance, which is reflected by morph's high OOV MTWV values, the highest achieved in the report. Despite this, the much lower IV MTWV of morph and combined systems containing it (which is most likely caused by a high false alarm rate) means that the net MTWV is worse than the word-based systems.

System	All	IV	OOV	Threshold
word	0.3987	0.5014	0	0.077
word-sys2	0.4033	0.5073	0	0.047
word + word-sys2	**0.4142**	**0.5209**	0	0.110
morph	0.3597	0.4296	0.08856	0.071
morph + word + word-sys2	0.3575	0.4259	**0.0922**	0.240

Table 3: System combination TWV scores when using the IBM WFST KWS systems, showing both individual performance and the performance when systems are combined.

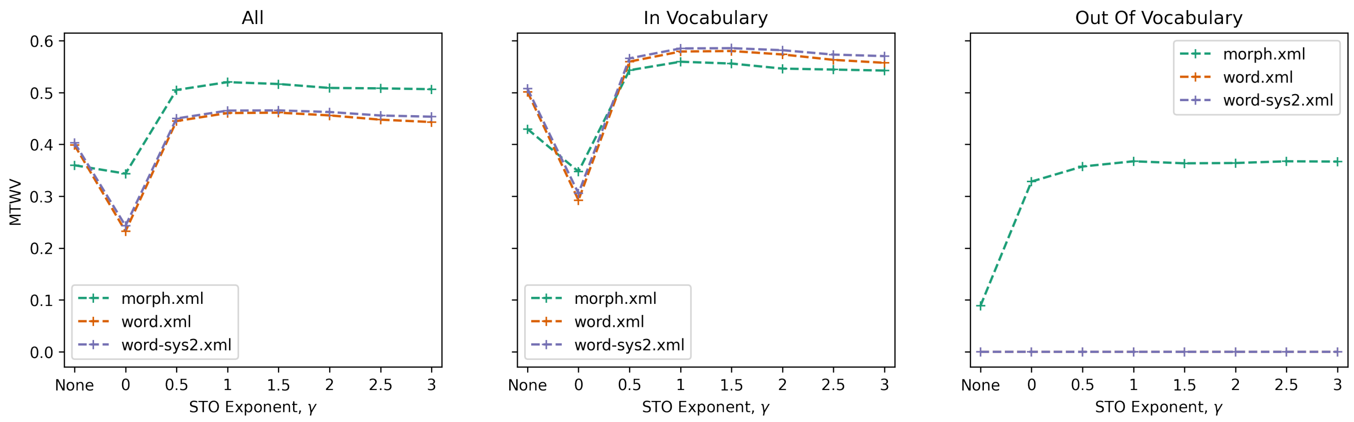

Figure 4: Plot of MTWV values for IV, OOV, and IV+OOV phrases as a function of the exponent in STO normalisation for the IBM WFST files word.xml, word-sys2.xml, and morph.xml. Score normalisation was applied by extracting the hits from each of the supplied IBM KWS system outputs and replicating the output except for using an adjusted score.

If score normalisation is applied to the combined IBM KWS systems then improvements to the MTWV values can be obtained. This is shown in Figure 4, where STO normalisation with $\gamma = 1$ is used, inspired by the good performance of $\gamma = 1$ in both non-morphed and morphed systems in Section 3. Now the best performing system is morph, as the OOV performance is very high (compared to the word-based systems which still have OOV MTWV = 0), and the IV MTWV is improved due to score normalisation, which polarises the hit scores and therefore selects only the most confident hits for each key word to reduce the probability of false alarm.

Figure 5 shows DET curves for two different combinations of the IBM systems after STO normalisation. What is interesting is how much poorer the ATWV values are compared to the MTWV, with scores even going negative[4]. The large discrepancy indicates that there is great sensitivity in the TWV values to the setting of the decision threshold θ, caused by the lattice systems picking up a large number of hits, and so allowing many more false alarms than 1-best systems if the decision threshold is too low. This can also be seen by the long tail in the DET curves, meaning that low miss probabilities are achieved, but only when the decision threshold is set to near 0, so at the expense of high probabilities of false alarm.

Despite this, the best MTWV values of this report were achieved by combining the STO normalised IBM lattice systems, with MTWV = 0.5443 for word + morph with $\gamma = 1$, and **MTWV = 0.5512 for word + word-sys2 + morph, again with $\gamma = 1$**.

[4]If a TWV is negative it means that $\beta P_{FA} > P_{Miss}$, so the sytem is less than β times as likely to detect a miss than a false alarm, which indicates a high false alarm probability, as normally we expect the probability of a miss to be more than βP_{FA}.

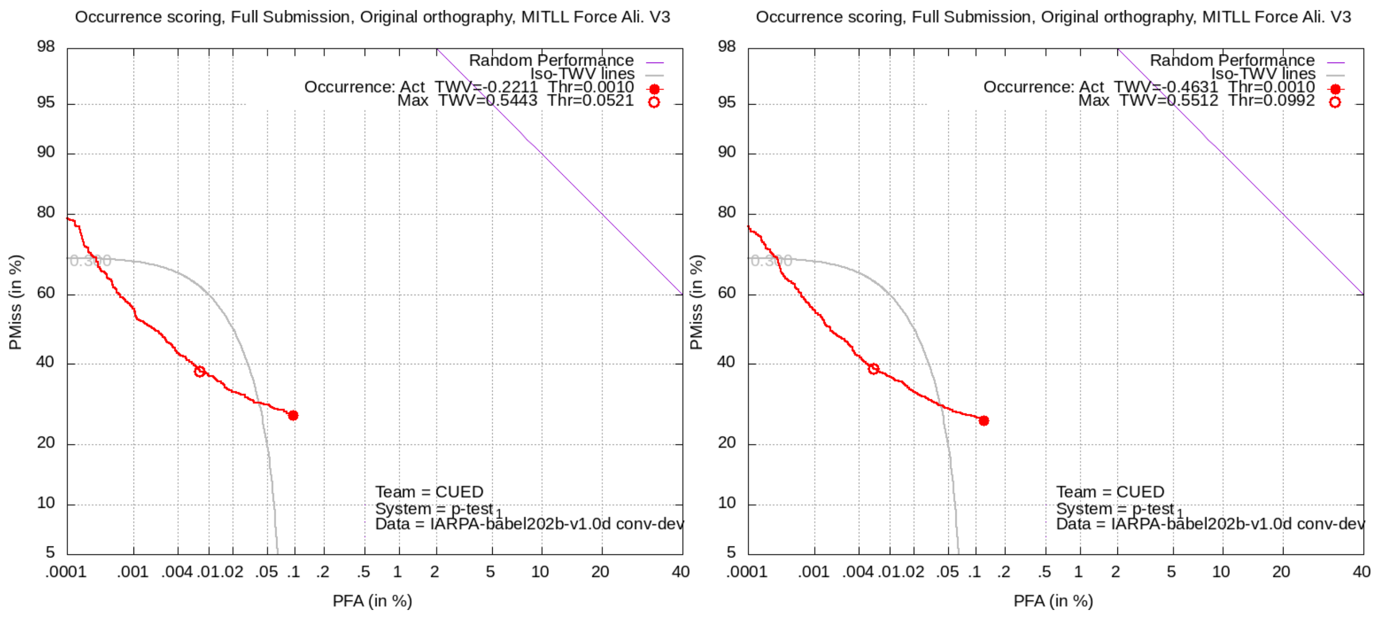

Figure 5: Plot of two DET curves, representing system combinations of STO normalised IBM WFST system outputs. The left is word + morph, each with $\gamma = 1$ for STO normalisation, and the right is word + word-sys2 + morph, again each with $\gamma = 1$ STO normalisation.

7 The Effect of Query Length

Finally, the effect of query length on MTWV was investigated for the morph.xml + STO ($\gamma = 1$) system, due it being the best performing *single* system investigated in the report. The lengths of queries ranged from 1 to 5, and the results are shown in Table 4 below. It can be seen that the queries containing two words contribute the most to the MTWV score, followed by single word queries. This could be because one word queries are more likely to be chosen as a hit due to them having fewer requirements such as duration between words being less than 0.5 seconds, which means there are a greater number of false alarms from the single word queries. In addition, longer queries have more opportunity to contain a mistake from the ASR system (as there are more words that can be erroneous) and therefore generate more misses than shorter queries. Therefore, the reason that two word queries perform best could be because they are the balanced middle-ground between these two issues, containing few enough words that the probability of them being a miss is lower, but great enough that their probability of being a false alarm is higher than one word queries.

Query Length	All MTWV	Percent of queries	Threshold
1	0.5257	59.0%	0.039
2	0.5747	34.4%	0.039
3	0.2930	4.1%	0.039
4	-0.0028	2.0%	0.039
5	0.0	0.41%	0.039

Table 4: Table of MTWV values on the morph.xml + STO ($\gamma = 1$) file, found via mapping each query in the query file to a specific length to generate the length.map file, and then selecting only queries with that length when running the scoring script (this is done by changing the KWS-term-map to length.map in termselect.sh, and the selection to the length key from that mapping).

8 Conclusion

This report investigated how smarter approaches to key word spotting can be used to improve hit statistics, such as MTWV, for low resource languages that have poor ASR system accuracy. The main conclusions of the report are as follows:

1. Morphological decomposition and grapheme mapping are able to improve performance of KWS systems by mapping OOV phrases in the query to morphs and IV phrases respectively, such that the OOV phrase may still present a hit if the ASR system had just made a mistake in the original classification.

2. Score normalisation, for example STO normalisation, can improve MTWV values by shifting/polarising the scores of occurrences in the ASR output to reduce the probability of false alarms.

3. Combining KWS systems in general improves performance, especially when STO normalisation is involved, due to the 'wisdom of crowds' effect of ensembles.

4. The WFST lattice-based KWS systems outperform the 1-best ones in both IV and OOV performance due to the greater amount of acoustic and phonological information they carry in them. However, they make setting a decision threshold more important as the high number of hits means that too low a threshold causes ATWV values to go negative due to many false alarms. The highest recorded MTWV was for word.xml + word-sys2.xml + morph.xml with $\gamma = 1$, and was MTWV = 0.5512.

5. Upon breaking down the hits down by query length, the highest MTWV contributions were from queries with two words in, likely due to this length being a good middle-ground between being too likely to be a false alarm (like one word queries) and being too likely to be a miss (like higher word queries).

References

[1] Jonathan Mamou, Bhuvana Ramabhadran, and Olivier Siohan. "Vocabulary Independent Spoken Term Detection". In: *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '07. Amsterdam, The Netherlands: Association for Computing Machinery, 2007, pp. 615–622. ISBN: 9781595935977. DOI: 10.1145/1277741.1277847. URL: https://doi.org/10.1145/1277741.1277847.

[2] Javier Tejedor et al. "A comparison of grapheme and phoneme-based units for Spanish spoken term detection". In: *Speech Communication* 50.11-12 (Nov. 2008), pp. 980–991. DOI: 10.1016/j.specom.2008.03.005. URL: https://doi.org/10.1016/j.specom.2008.03.005.

[3] *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013*. IEEE, 2013. URL: http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=6619549.

[4] Damianos Karakos et al. "Score normalization and system combination for improved keyword spotting". In: *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*. 2013, pp. 210–215. DOI: 10.1109/ASRU.2013.6707731.

[5] Steven Wegmann et al. "The TAO of ATWV: Probing the mysteries of keyword search performance". In: *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*. 2013, pp. 192–197. DOI: 10.1109/ASRU.2013.6707728.

[6] Ilyes Rebai, Yassine BenAyed, and Walid Mahdi. "A novel keyword rescoring method for improved spoken keyword spotting". In: *Procedia Computer Science* 126 (2018), pp. 312–320. DOI: 10.1016/j.procs.2018.07.265. URL: https://doi.org/10.1016/j.procs.2018.07.265.

[7] Dan Jurafsky. *Minimum Edit Distance - Stanford University*. URL: https://web.stanford.edu/class/cs124/lec/med.pdf.