Hadith Segmentation Using N-grams
Introduction to NLP (EECE 634) Project Report

Ahmad Mustapha,[1] Joseph Sabbagh[2]

[1]Electric and Computer Engineering, American University of Beirut
[2]Computer and Communication Engineering, American University of Beirut

In this project report, we aim to showcase a Natural Language Processing (NLP) technique that aims to segment hadith texts into its main components: Sanad, and Maten. After collecting approx. 16,000 hadith with Sanad Maten annotations from 8 hadith books we build a segmentation system achieving 64% accuracy. The entire data process that leads to the final product i.e. data collection, data model, fine-tuning, and evaluation are described below.

Introduction

Islamic ahadith are historical narrations that describe the acts and statements of Prophet Mohammad (PBUH), his household, and his Companions. Those ahadith are collected and compiled into different books by early Islamic scholars. They are still enjoying the attention and are of much importance for historical/Islamic studies. And they are being by Islamic scholars to extract the sharia law as the Holy Quran is not the only resource for Islamic laws. Figure 1 shows a hadith example.

Throughout the years, Islamic hadith collectors have developed a unique structure to record hadith. A single hadith is mainly composed of two parts:

1. Sanad (Arabic.السند): Which lists the human channel in which the hadith was propagated and is usually called *the narrators chain*. In other words, it mentions that person x told person y ... that he heard person z said.

2. Maten (Arabic. المتن): Which contain the body of the hadith/statement original sayer (in this project we targeted Mohammad (PBUH) hadith only) that the narrators propagated.

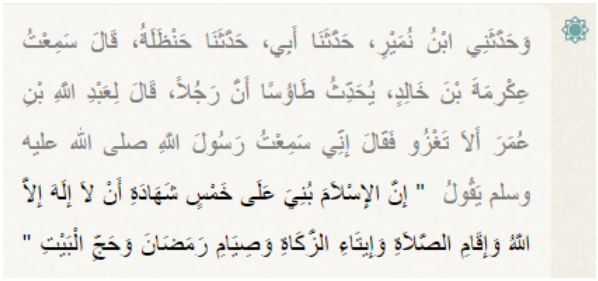

Figure 1: A regular hadith structure where the greyed out part represents a sanad with narrator chain visible and the black part represents a maten.

As you can see in hadith books hadith usually is represented as one text without the separation between its part. Segmenting those parts have it's importance in the research historical/Islamic research community. As some of them want only to focus on sanad to extract the narrations chains and visualize them or study them. While others are interested in Maten only. Here comes the motivation for our work.

In this work, we collected 16,000 annotated hadith and we used part of them as training data to extract useful features to discriminate between Sanad and Maten parts. The NLP technique we used is adopted from (*1*) but with some enhancements from our side. Link to the code repository [1] .

The rest of the report is structured as follows: Section 1 describes the data collection procedure and output. Section 2 describes the hadith segmenter model used. Section 3 defines the evaluation metric used and shows experimental results. We finally conclude with a conclusion.

[1]https://github.com/AhmadM-DL/Hadith-Segmentation-Using-Ngrams

1 Data Set Description

To formulate a good text segmentation model we needed to have a considerable amount of annotated hadith. Such data set was not openly available and we had to find alternatives. The below subsections describes the data pipeline.

1.1 Data Collection

We found that Sunnah.com lists several hadith books and presents them in a very structured manner. We decided to scrape the website. Though we find later that the website acknowledged in its policies that it doesn't accept scraping its content - hence we refined from posting the scraped dataset publicly. We scraped 8 books. For each book, we scraped all available hadith in all volumes and chapters along with the book title and description. Some of the hadith on the website doesn't have an annotated sanad yet we scraped them and discarded them in the following stages of the pipeline. Table 1 lists information about each scraped book.

Book	All Hadith	Hadith with sanad	# word per sanad			# word per maten		
			min	avg	max	min	avg	max
Jami` at-Tirmidhi	4153	1912	6	34	160	1	16	464
Muwatta Malik	1986	428	12	34	238	1	14	88
Riyad as-Salihin	1895	933	5	19	366	1	17	519
Sahih Muslim	7378	2999	11	41	372	1	15	538
Sahih al-Bukhari	7061	2910	2	38	349	1	17	477
Sunan Abi Dawud	4840	1992	1	38	211	1	13	153
Sunan Ibn Majah	4136	2275	9	38	185	1	14	191
Sunan an-Nasa'i	5691	2531	16	39	228	1	13	426
Total	37140	15980						

Table 1: Collected Hadith Info.

1.2 Data Pre-processing

1.2.1 Cleaning

The hadith extracted form the website is mostly clean. However, some of them are not. For example, some of the hadith contained HTML. To deal with this problem we had to remove all non-Arabic characters whether numbers or alphabets from extracted hadith. We also removed all punctuation, special uni-codes such as left-to-right and right-to-left marks. We Finally removed the Arabic diacritics and all possible tashkeel.

1.2.2 Structuring

Originally the books were structured and saved as JSON files. As we are only interested in sanad/maten lists of hadith, we removed all hadith that doesn't contain sanad and then we collected all 15980 sanad/maten couples from all books in one large list. This list is then split at random as 70% Train Set (11,186 hadith) and 30% Test Set (4,794 hadith). The Train set is fed into a script that generates precompiled sanad/maten bi-gram and uni-gram lists. And the Test set is fed into scripts that evaluate the model performance. The Test Set contains a list of tuples that each contains three elements. the first element is the entire un-split hadith that the model has to predict. The second and the third elements are the ground truth split token and the ground truth split position.

2 NLP Technique

In this section, we describe the segmentation model used. As aforementioned our mode algorithmic approach is adopted from (*1*). Figure **??** shows the model overview. The idea is simple. When a new hadith is fed to the model, it is pre-processed by removing tashkeel, white spaces, and punctuation. After that, it is tokenized into 2-grams and each token is checked whether it appeared in a precompiled Maten token list or Sanad one and then the token is labeled accord-

ingly as ”MATEN” or ”SANAD”. If the token didn’t belong to any of those it is labeled as ”UNCERTAIN”. After labeling, the segmenter finds the best split that signals the end of the sanad and the start of maten. We found that the author’s splitting approach doesn’t work well with our ground truth so we used another approach.

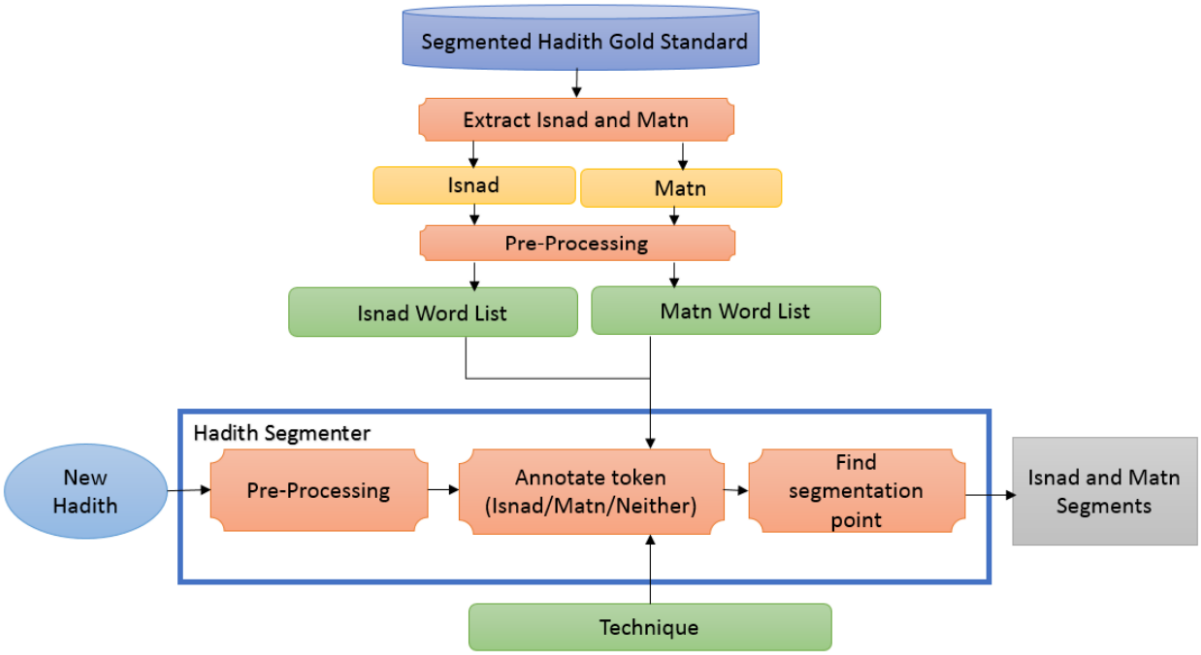

Figure 2: Hadith Segmenter Model

2.1 Original Technique

The initial procedure used by the authors to label hadith tokens is shown in algorithm 1 below. The backing off procedure to uni-grams is used to overcome the problem of missing information and irregularities. For example consider the tri-gram token ”سعيد بن عباس” that didn’t appear as it is in precompiled lists. This will lead to the annotation of the token as ”UNCERTAIN” while it is a sanad token with high probability. Breaking off the token into bi-grams will make the entire token a sanad one as ”سعيد بن” and ”بن عباس” both of them appeared in the sanad bi-gram precompiled list.

However, the authors ended up using only bi-grams and uni-grams lists as it turned out to be much useful. And this what we considered also.

In order to split the hadith tokens sequence, the authors formulated a simple technique. The technique passes over the tokens and when it reads two consecutive not-sanad tokens it signals the first token as the split position.

When we applied this approach the model performance was very bad. We will call this approach the "baseline" approach. One question hit is why the approach worked for the authors but not for us. First, the authors extracted pre-compiled sanad/maten n-grams from 40 diverse hadith only and tested on another 200. Second, there is an important difference between their target task and ours. This difference arose from the ambiguous definition of sanad/maten. In their definition sanad is the part of the text that only includes the narrator chain and everything else is a maten. In our definition sanad is the part of the text that precedes Mohammad's (PBUH) statement. In hadith of Regular type both definition coincides as the narrators-chain is directly followed by Mohammad's (PBUH) statement. However for other types of hadith ex. Introductory ones there is a text that sets between

Algorithm 1 Annotate Tri-gram tokens

Tokenize Hadith into Tr-igrams T
for $t \in T$ **do**
  if $t \in IsnadTrigramList$ **then**
    $Label\ t$ **Isnad**
  else if $t \in MatnTrigramList$ **then**
    $Label\ t$ **Matn**
  else
    Convert t to Bigrams b
    if $b \in IsnadBigramList$ **then**
      $Label\ t$ **Isnad**
    else if $b \in MatnBigramList$ **then**
      $Label\ t$ **Matn**
    else
      Convert t to Unitgram u
      if $u \in IsnadUnigramList$ **then**
        $Label\ t$ **Isnad**
      else if $u \in MatnUnigramList$ **then**
        $Label\ t$ **Matn**
      else
        $Label\ t$ **Niether**

Output:

Token1	Lable1	Token2	Label2	...

Figure 3:

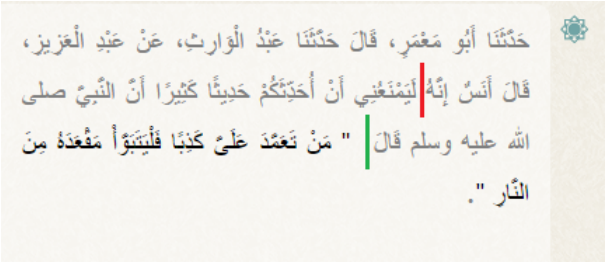

Figure 4: An introductory hadith showing the prediction difference between our split predictor (green) and the baseline predictor (red). The greyed text represents sanad ground truth. The text between the two predictions is neither a narrator chain nor a statement

the narrators-chain and Mohammad's (PBUH) statement. show how we align in the definition of sanad in regular hadith and how we diverge in other types of hadith.

2.2 Our Modification

The "baseline" splitting approach would never work to split hadith text according to our definition of sanad and maten. To tackle the problem we formulated another split position predictor. The predictor made use of the Information Gain algorithm to detect the best split. For a given labeled tokenized hadith we try all possible splits and we choose the one that returns the highest Information Gain ratio. Figure 4 shows a prediction difference between the baseline and the information gain predictor.

2.3 Precompiled Lists Generation

In this section, we present our approach to generating precompiled maten/sanad n-grams lists from the Training Set. Those lists act as lexicons for both hadith parts sanad and maten. To generate those set we first extracted all bi-grams and uni-grams for each sanad and maten hadith parts. We then extracted the unique ones. For the sanad lists, we took the most frequent n-grams only while for maten we took all the possible unique n-grams. The top frequent percent is a parameter in our approach we will denote it by $TF\%$. During the fine-tuning phase, we

tried different values for the top frequent percent until we found the optimal results as shown in section 3

3 Evaluation

To evaluate the model performance we formulated the following metric. Let H be the total number of hadith in the Test Set. Let the ground truth split position of a hadith be denoted as $spos$ and the predicted one as $\hat{spos}$. We also define the function denoting correctly predicted hadith $Tpred(hadith)$ using the indicator function $1(abs(spos - \hat{spos}) <= n)$ and $Fpred(hadith)$ to be $1 - Tpred(hadith)$. To this end we define our k-tolerated accuracy metric as follows:

$$KTACC = \frac{\sum_{h=0}^{H} Tpred(h)}{\sum_{h=0}^{H} Tpred(h) + \sum_{h=0}^{H} Fpred(h)}$$

This accuracy denotes how much of the testing data predicted split positions have been far by K tokens from their corresponding ground truth split position. In the below section we show the experiments we do by varying $TF\%$ and measuring $KTACC$ accordingly.

3.1 Fine Tuning

We did more than 20 experiments while using different data sets and slightly different approaches. 13 of those experiments were held on the following books [Jami‘ at-Tirmidhi, Sahih Muslim, Sahih al-Bukhari, Sunan Abi Dawud, Sunan an-Nasa’i, Muwatta Malik]. During these experiments, we tested different top frequency percent $TP\%$ parameter ranging from 0.25 to 7. For each of those, we tried the baseline predictor and the information gain predictor. The results of these experiments are shown in figure 5.

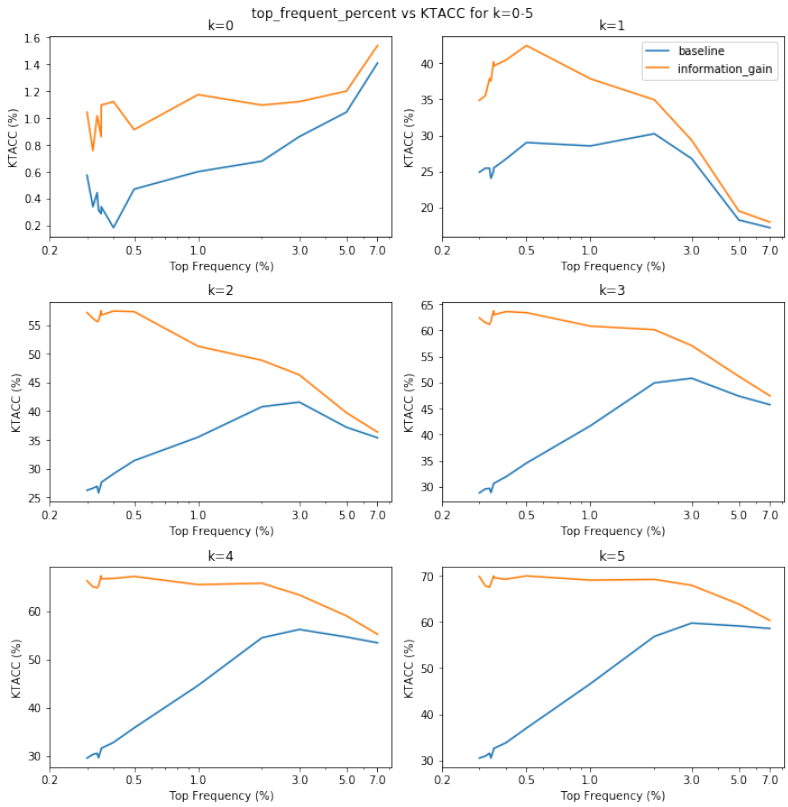

Figure 5: The results of 13 experiments fine tuning $TP\%$ parameter.

As you can see from the figure the information gain approach always topped the baseline approach. For 0-TACC the prediction accuracy for both predictors was near zero. This means only about 38 hadith out of 3800 (not the full data set this data set used only 6 books) hadith have been exactly predicted.

There is an obvious pattern that as we choose higher $TF\%$ (i.e. take less frequent n-grams) the lesser the accuracy is for all k's except for 0 (which can be neglected for the very small accuracy anyway). This makes sense because as we take less frequent n-grams in the sanad list the more they will overlap with the one in the maten list leading to very noisy labeling. On the other hand, decreasing $TF\%$ to 0.3% represents the maximum accuracy throughout all the experiments.

AS we are more interested in small k experiments (4-TACC can't be considered a good estimator of accuracy because hitting the prediction with more than 4 tokens apart from the ground truth is not use full for any task) we can focus on the 3-TACC curves. As we can see the information gain curve peaked at $TF\%$ = 3% reaching 63.75% 3-TACC accuracy which means that 2442 out of 3800 were split with three or fewer tokens apart from the ground truth token.

To push the model more we added two more books to our data set. This time the highest possible 3-TACC was 64.6%. Thus it is obvious that this is the maximum capacity of the model and this is how far using simple n-grams lexicons can take us in our task. This makes sense as to figure the split in non-regular hadith requires some contextual information for each word. This can be achieved by using attention-based models. This might be a good future work.

One last thing to mention is that it seems that our corpus is composed of 30% regular hadith those are captured by the baseline, the other 30% is captured by our split predictor and are composed of non-regular hadith with a number of tokens between the narrator chain and the statement is moderate. The other 40% is composed mostly of non-regular hadith with the distance between the statement and the narrator chain is wide.

4 Conclusion

In this paper, we showed case an NLP technique that aims to detect the split between a hadith text parts sanad and maten. the results showed that the technique can predict regular and easy non-regular hadith but fails to predict complex and harder ones. Other types of contextual NLP models should be able to deal with those types of hadith.

References and Notes

1. S. Altammami, E. Atwell, A. Alsalka, *Proceedings of the 3rd Workshop on Arabic Corpus Linguistics* (2019), pp. 31–39.