Tracing Traditions: Automatic Extraction of Isnads from Classical Arabic Texts

Ryan Muther and David Smith
Northeastern University
Boston, MA
{muther.r, davi.smith}@northeastern.edu

Abstract

We present our work on automatically detecting *isnads*, the chains of authorities for a report that serve as citations in hadith and other classical Arabic texts. We experiment with both sequence labeling methods for identifying isnads in a single pass and a hybrid "retrieve-and-tag" approach, in which a retrieval model first identifies portions of the text that are likely to contain start points for *isnads*, then a sequence labeling model identifies the exact starting locations within these much smaller retrieved text chunks. We find that the usefulness of full-document sequence to sequence models is limited due to memory limitations and the ineffectiveness of such models at modeling very long documents. We conclude by sketching future improvements on the tagging task and more in-depth analysis of the people and relationships involved in the social network that influenced the evolution of the written tradition over time.

1 Introduction

In classical Arabic texts, lists of the names of authorities that transmitted a piece of information (*isnads*) are often attached to a statement or report (the *matn)* to confirm its reliability. The study of *isnads* is an integral part of the study of *hadith* and the history of the Arabic written tradition in general. With the increasing availability of digitized texts, new methods are required to automatically locate and analyze *isnads* at scale in a wider variety of text than the canonical *hadith* collections used in smaller-scale studies (Harrag et al, 2014; Maraoui et al, 2019, Altammami et al, 2019). *Isnads* are often seamlessly integrated into running text, and are therefore difficult to distinguish from the surrounding text based on visual layout or punctuation information alone. The textual content of the *isnad* itself must therefore be used to determine its location. For instance, in the example *hadith* below, the names and transmissive terms indicate that the underlined section in the beginning of the text is the *isnad,* while the remainder is the *matn*.

حدثنا أبو داود قال: حدثنا هشام، عن قتادة، عن الحسن عن سمرة، أن النبي صلى الله عليه وسلم: قال
من قتل عبده قتلناه ومن جدعه جدعناه ومن خصاه خصيناه

"Abū Dāwūd transmitted to us, he said, 'Hishām transmitted to us, from Qatādah, from al-Ḥasan, from Samurah that the Prophet, may the peace and blessing of God be on him, said, 'Whoever kills his slave, we will kill; and whoever mutilates, we will mutilate him; and whoever emasculates, we will emasculate him.'" (Sulaymān, 204AH)

In theory, one could also extract the body of the *hadith,* but there are fewer significant linguistic cues that set a *matn* apart from the background text than there are for its corresponding *isnad*. Once one has

Proceedings of the Fifth Arabic Natural Language Processing Workshop, pages 130–138
Barcelona, Spain (Online), December 12, 2020

amassed a large collection of *isnads*, one could begin to extract the relationships between individual transmitters, as well as infer the exact identities of transmitters given the relationship network's structure and the variety of names used for the same individual. This would allow humanists to draw a larger scale picture of the evolution of the Arabic written tradition than has previously been possible by analyzing the social network involved in both the creation of individual texts and the corpus as a whole.

Long documents often contain passages in an *embedded genre*, distinct from the surrounding text. We can locate certain kinds of embedded texts using visual layout information to segment the text into smaller units, like articles in a newspaper, which could then be classified using any number of text classification methods (Lee et al, 2020), or using formatting and punctuation cues to locate poetic passages embedded in prose works (Lorang et al., 2015; Foley, 2019). If one does not have the ability to divide the text up into sufficiently granular units, or the embedded genre of interest is not easily separable from the surrounding background text using layout cues alone (e.g., indented lines of poetry), these approaches are not possible. However, *isnads* tend to have linguistic cues that indicate their presence, like the presence of a large number of names or particular transmissive terms in a small region of the text. In this paper we will follow a two-step approach to solving the problem of identifying *isnads* in long documents by first applying a retrieval model to find sections of a document likely to contain start points of *isnads*, then training a sequence tagging model on the retrieved sections to identify where within those spans the *isnads* begin. Upcoming work will involve inferring the endpoints of the *isnads* given the start points identified by the tagging model using a span prediction model.

This paper is organized as follows; in section 2 we present an overview of related work. Section 3 provides a deeper discussion of the data sources we use for our experiments. In section 4, we present the results of an inter-annotator agreement study done as part of the process of creating training data as a benchmark for human performance on the task. Section 5 discusses the models used in our experiments, as well as hyperparameters and training processes. Section 6 presents the results of our evaluation. Section 7 presents possible avenues of future work.

2 Related Work

This work is closely related to the problem of named entity recognition. Most named entity recognition models are evaluated using datasets like the CoNLL 2003 dataset (Tjong and de Meulder, 2003). In contrast to our data, the target entities in such datasets are at most a few words long, the training and test documents are much smaller than the whole texts we are working with, and almost all documents contain named entities. Neural models such as the BiLSTM and BiLSTM-CRF, as used by (Lample et al, 2016) have been shown to perform well on a performing named entity recognition in multiple languages. Additionally, a great deal of effort has been expended to improve the performance of named entity recognition on short documents like those found on social media sites like Twitter (e.g., Ritter, 2011). We seek to do the opposite, and explore performance on longer documents. While we could have approached the tasks described above as named entity recognition tasks, looking for names in *isnads* in particular, downstream tasks like information extraction or network inference would likely benefit from the additional structure present in the text around the names.

Unlike prior work on automatic *ḥadith* tagging, we focus exclusively on trying to identify *isnads*, rather than also trying to simultaneously extract the corresponding *matn* for each chain (Harrag et al, 2014; Maraoui et al, 2019, Altammami et al, 2019) or extract information from the identified isnads (Siddiqui et al, 2014), as others have done. We are working with a much larger and correspondingly more diverse collection of texts, rather than limiting our analysis to *hadith* collections. Finally, much of the existing work, like that cited above, focuses on using rule-based systems to identify isnads and the individual transmitters within them which limits the generalizability of these models to previously unseen texts.

3 Data and Preprocessing

The data for these experiments comes the Open Islamicate Texts Initiative (OpenITI) corpus (Romanov and Seydi, 2019,) a collection of 4,285 transcribed texts in classical Arabic collected from digital libraries by scholars from the University of Vienna and University of Leipzig totaling 1.5 billion words. The individual documents in the OpenITI corpus are complete texts and are often quite long, with the longest single text containing over 112 million words. The texts in the corpus are largely unpunctuated,

and what punctuation there is is a modern editorial intervention, so we have removed punctuation from the texts when it exists.[2] Additionally, training any models on punctuated text would harm the performance on the model on the unpunctuated texts that make up the majority of the corpus. Since most of the corpus is unpunctuated, it is difficult to break the texts down into smaller units than complete texts. We have also performed orthographic normalization to remove different variants of the same character so that the model is not influenced by the orthographic choices of any particular author.

The training dataset we have created consist of data from fifty labeled texts, with 163 distinct tagged regions of text. While *isnads* are most common in *hadith* collections, they are also commonly used in historical writing, exegesis (*tafsir)*, geographies, and literature. The specific texts were chosen over the course of several rounds of annotation, in which a CRF was trained and tested on existing training data, then the resultant tags in were analyzed by scholars to see where the model tended to fail to properly label *isnads*. When a weakness was found, new texts were selected that, in the experts' opinion, contain examples that could be used to give the model an understanding of how to avoid such failures in the future. In total, the tagged text consists of 907,110 tokens containing 3,071 *isnads*. The average length of the isnads in the tagged data is 31.9 tokens.

4 Inter-Annotator Agreement Study

To create training data for this task, we have used an iterative process of human-in-the-loop model training. Working in conjunction with experts in the fields of Islamic history and religious studies with a strong command of classical Arabic, we began by annotating a single text. Using these initial annotations, a CRF with token features (see section 5) was trained and used to automatically annotate other texts, which were then corrected by the annotators, creating additional training data while being able to use the automatic tags as a starting point. This not only makes annotation easier, as it presents the annotators with the task of modifying pre-existing tags created by the model rather than inserting tags from scratch, but also can be used to create training data that specifically addresses failures by the previous version of the model. As an illustrative example, one version of the model would often mislabel lists of students in a scholar's biography as chains of transmission, likely due to the presence of a high number of names in those regions of the text and a lack of examples to the contrary which would encourage the model not to label such sections as chains of transmission. This annotation process allowed the annotators to recognize that particular kind of language as a failure point of the model and create new training data which the model can use as evidence that those sorts of linguistic constructs are not chains of transmission. If the human annotators did not perform this process with the initial output from the model, it is possible that their annotations would not catch these sorts of cases, resulting in a less useful training dataset. By specifically providing examples of the kinds of data that confuse the model, we end up with a model whose judgements more closely resemble those of humans over several iterations, both as a result of the increased training data and the careful selection of the new text to annotate.

To get some understanding of how difficult the task of labeling *isnads* actually is for humans, we performed an inter-annotator agreement study using data from five of the annotators involved, each of whom annotated the same text, totaling 2000 lines of text across different sections of one work. Token level agreement between individual annotators is reported in Figure 1a, while the variance in the locations of start and end points of overlapping labeled spans is shown in Figure 1b. Overall, we see high agreement at the token level, which indicates that the task is not overly difficult for expert annotators, with Krippendorf's α of 0.84 indicating very strong agreement across all raters. Additionally, when multiple annotators marked overlapping spans, they all agreed on the starting point of the span seventy percent of the time, while the location of the endpoints varied more often, with complete agreement among the annotators occurring only fifty-three percent of the time. However, the vast majority of the disagreements were very small. The outliers in figure 1b were found to be mostly cases where annotators merged two adjacent *isnads*. This is a rare occurrence in the training data, with around three percent of identified *isnads* beginning right after the end of an *isnad*. There are two main points to take from this annotation when evaluating the results of any model attempting to solve this task. First,

[2] Similar to how Latin was largely unpunctuated prior to the introduction of printing, with verbs acting as sentence end markers, in classical Arabic certain words or phrases indicate transitions between sentences and clauses in lieu of punctuation. For more information, see https://www.britannica.com/topic/punctuation/Punctuation-in-Asian-and-African-languages

that identifying the end points is more difficult then identifying the start points for this particular task, and second, that properly segmenting adjacent spans is difficult even for human annotators.

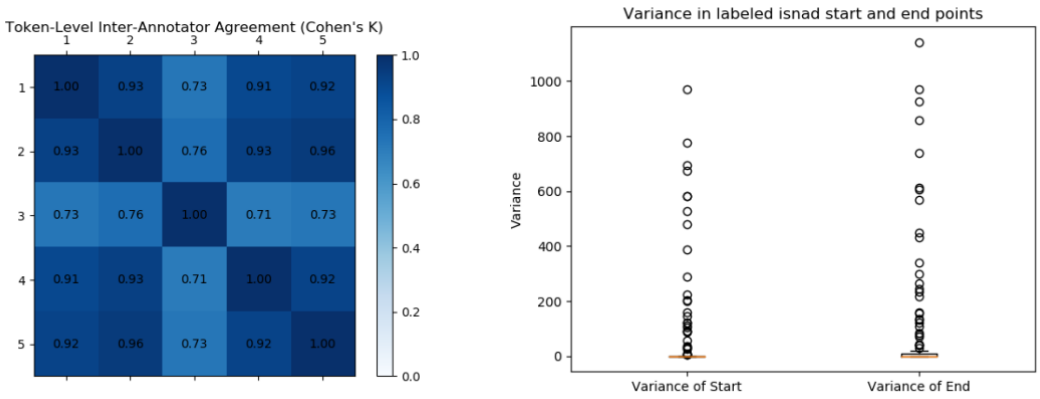

Figure 1. Inter-annotator agreement (1a, at left) and variance (1b, at right, in tokens) statistics

5 Experiments

5.1 Sequence Tagging Models

We initially experimented with solving this task by sequence labeling alone. These models tag every token in an input document as either the beginning of an *isnad* ("B"), inside an *isnad* ("I") or outside an *isnad* ("O"), as would be done for tasks like named entity recognition. As a baseline for comparison, we train CRF models (Lafferty et al., 2001) using counts within a fixed-size window on either side of the word to be tagged as well as the token to be tagged itself as features for the model. Thus, if the word "the" occurred twice before the token being tagged, the feature "the_before" would be created with the value 2. We also investigated using unsupported features (features for words that do not occur), but this created a large number of features that made model training too memory-intensive. The intuition behind these features is that the spans of text we seek to identify share common language that these features will enable the model to learn. Additionally, this form of model is able to use negative examples provided by the annotation to learn how the context of a word affects its likelihood of membership in a target span. Using information about the words around the word it is trying to tag, the model is able to learn how the context of a word affects its meaning, and not naively assume that all instances of the same word should be treated equally. One might try to solve this problem by training a pair of language models using example isnads to train an *isnad*-specific model and a general background model. Using these two models, one could attempt to label entire spans using the ratio of log odds between the two models to decide how to segment a text into *isnad* and background sections using a Viterbi-like inference method rather than making decisions at the level of individual tokens. However, such a model's inability to understand context would result in a model with no ability to determine when common words in the embedded genre should and should not be treated as part of one, giving many false positives as single words common in the embedded genre would often be labeled as part of one, regardless of their context.

We also train CRFs using 200-dimensional GloVe embeddings trained on the complete OpenITI corpus using the default parameters (Pennington et al, 2014) as features, concatenating the embeddings of the words within a window around the token to be tagged and using that as input to the model. Since different authors have different styles of citation, using different terms to indicate transmission, and draw from different sources, thus including different names in their chains, the unseen texts in the test set will contain chains that have a distinct vocabulary from those seen by the model in training. For the surface level token features described above, such mismatches in vocabulary are tokens that the model has no ability to understand the meaning of, as it never encountered them in training. Using the distributed representation provided by the GloVe embeddings, the model can leverage information about tokens with similar distributional semantics (i.e. other names or transmissive terms with similar embeddings) and gain some understanding of the meaning of an unknown token through its similarity to previously-seen ones, improving the generalizability of the model to previously unseen texts.

Additionally, we train a BiLSTM (Hochreiter and Schmidhuber, 1997; Graves, 2005) with a hidden layer size of 100 using 200-dimensional word embeddings fitted to each of the corpora, as discussed above. To speed up training, make processing long documents feasible, and avoid the known issues LSTMs have when modeling long documents, we split the documents in the OpenITI corpus into chunks of no more than 1000 words which are then used to train the model. After training, the results for the chunks are then concatenated to get results for the complete document. While this may cause issues when the chunking process splits the target spans into two pieces, the length of the chunks can be chosen such that the risk of that occurring is minimized. Furthermore, the chunks from within a document are padded with special symbols distinct from those used at the beginning and end of a document, allowing the model to learn that certain chunk beginnings are more likely to be part of the embedded genre, despite the lack of starting context provided by the beginning of the span. Although such splits are not impossible to recover from with sufficient training, they still present a nontrivial issue. This issue, in conjunction with memory limitations related to tagging very long texts in caused us to shift focus and move from trying to detect the spans with a single model to a two-step retrieve-and-tag approach to only detect start points, eliminating both the computational expense of modeling very long texts using these sequence to sequence models and the possibility of missing an isnad due to splitting it across chunks.

5.2 Retrieval-Tagging Models

If we are going to split the documents into chunks, it is worth noting that only 30% of the 100-token chunks contain starting points of *isnads*. Consequently, it may be beneficial to first filter out chunks that are unlikely to contain *isnad* starting points. The first step in the process of finding the starting points of is that of document retrieval. The two-stage retrieval-tagging models are trained as follows. First, the corpus is divided into training, validation, and test sets, where the retrieved test documents will ultimately be used to evaluate performance on the start point labeling task. In order to train the retrieval model, for which we use a bag-of-words logistic regression model, the test documents are divided into two halves at random and each document in both halves is divided into chunks of at most some fixed size, we experiment with the sizes {25, 50, 75, 100}. The chunks in each half of the training set are used to train a retrieval model to retrieve documents containing *isnad* start points which is used to retrieve documents in the other half of the training set. The union of these two sets of retrieved documents are used as training data for the tagging model. The whole training dataset is then used to train a retrieval model to retrieve chunks from the test data. Using the complete set of retrieved chunks from the training data and the retrieved chunks from the test data, we then train a single layer LSTM with hidden layer size of 100 to tag tokens as beginning an *isnad* or not using 200-dimensional GloVe embeddings as our starting token representations.

For the retrieval model, we experiment with a variety of different parameter settings to obtain a high upper bound on the recall of the overall model, which we measure through an oracle experiment in which all retrieved chunks are tagged perfectly. Table 1 below presents an overview of the different feature sets and chunk sizes used. We experiment with unigram, bigram, and trigram features, as well as choosing to weight the features by TFIDF scores or not. As we can see, longer documents tend to be easier to retrieve, while bigrams tend to give the best performance, and TFIDF scoring significantly improves performance. It is possible that informative trigram features generalize less well across texts and between authors due to variations in the sources, and thus in the names that occur in *isnads*, resulting in lower retrieval performance.

Features	Chunk Size	Recall
Unigram	25	.767
Unigram	50	.848
Unigram	75	.872
Unigram	100	.860
Bigram	25	.802
Bigram	50	.857
Bigram	75	.887
Bigram	100	.849
Trigram	25	.784
Trigram	50	.802
Trigram	75	.864
Trigram	100	.874

Features	Chunk Size	Recall
Unigram, TFIDF	25	.854
Unigram, TFIDF	50	.886
Unigram, TFIDF	75	.896
Unigram, TFIDF	100	.941
Bigram, TFIDF	25	.913
Bigram, TFIDF	50	.915
Bigram, TFIDF	75	.959
Bigram, TFIDF	100	.935
Trigram, TFIDF	25	.896
Trigram, TFIDF	50	.943
Trigram, TFIDF	75	.954
Trigram, TFIDF	100	.939

Table 1. Recall scores for retrieval models using various feature sets and chunk sizes with an oracle tagger.

Additionally, for each model, we select a retrieval threshold which optimizes the F2 (recall-weighted F-score) of the resultant model. The default threshold of 0.5 tended to have comparatively low recall at the chunk level, as can be seen in Figure 2, which presents an example precision-recall curve for a retrieval model with 2-gram, TFIDF-weighted features and a chunk size of 75. Note that chunk level recall (the fraction of chunks that contain *isnad* beginnings that are retrieved, rather than the fraction of *isnad* beginnings that are retrieved) is only around 0.2 with the default retrieval threshold of 0.5, while if a threshold around 0.15 were used, recall would be around 0.95 at the cost of only a small drop in precision. This has the added advantage of adding the "attractive distractors" that were mislabeled by the retrieval model to the training and test set for the tagging model, while those chunks that, from the retrieval model's perspective, clearly do not contain the starting point of an isnad are excluded. This results in training and test sets that do not contain the very easy examples that the retrieval model can dismiss, focusing the training data on the more informative examples. Other parameter settings exhibit this same trend, albeit with different optimal operating points and upper bounds on recall. For the later experiments with *isnad* start labeling, we will use a bigram retrieval model with TFIDF scores and a chunk size of 75, as it gives the highest upper bound on recall in this corpus, with an optimal threshold of 0.14.

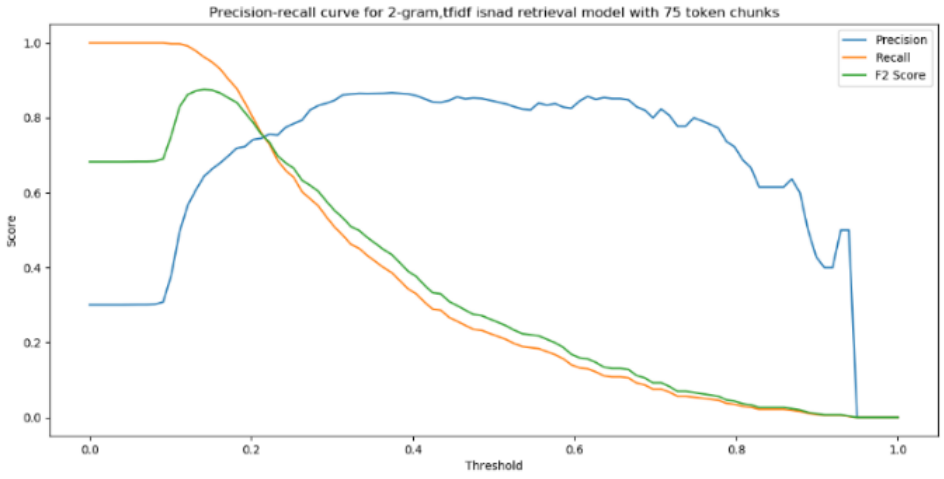

Figure 2: Example precision-recall curve for an *isnad* beginning retrieval model

6 Evaluation

6.1 Tagging Results

The token-based CRF, in addition to detecting start points, also labels tokens as part of an *isnad* using the "I" tag, as noted above. As such, we can evaluate not just the ability of the model the find starting point, but at identifying entire isnads. It should be noted that all results are given for models using eighty percent of the data for training, and ten percent each as validation and test data. The results for this task can be seen in Table 2 below.

Exact Precision	Exact Recall	Exact F1	Partial Precision	Partial Recall	Partial F1
.270	.241	.254	.873	.804	.835

Table 2: Span Level Results for the Token-Based CRF Sequence Labeling Model

The exact metrics report the precision, recall, and F1 when spans are considered correct if the model correctly identifies both the start and endpoints of an *isnad*, while the partial metrics give credit for an *isnad* if the model labels part of it as an *isnad*, even if the start and endpoints are incorrect.

6.2 Retrieval Results

We will focus on the results for the retrieve-and-tag approach to solving the problem of locating *isnad* start points, as seen in Table 3 below. As a baseline for comparison, we will compare our model to a naïve model which assumes that all instances of common *isnad* beginning terms, begin *isnads*.[3] We also compare the hybrid retrieval-tagging model to a CRF with local token features (as described in section 5.1) We report only the token level scores for the "Begin" tag, even though the model does solve the full problem of determining both where isnads begin and end. The retrieval portion of the combined model is, as stated above, a bigram bag of ngrams model trained on documents of 75 tokens in length using TFIDF scores as feature weights. All results shown are micro-averages across ten training/test splits at the document level (i.e. all the chunks of a tagged section of text are either all in the training set or all in the test set.)

Model	Precision	Recall	F1
Naïve	.188	.378	.219
CRF (Tokens)	.475	.422	.432
Retrieval-LSTM	.571	.442	.476

Table 3: Results for Isnad Beginning Labeling

While this model clearly outperforms both the naïve baseline and the CRF with token features at detecting isnad starting points, more will need to be done to improve its performance before moving on to the problem of finding the endpoint associated with a given starting point in the retrieve-and-tag framework.

7 Future Work

As noted above, the most immediate direction for future research is finishing the process of locating complete *isnads* in the hybrid model regime. Additional work will also be done to improve the performance of the *isnad* start point tagging model. It may be that additional features such as a gazetteer of common transmissive terms or the output of a named entity recognition system may improve the model's ability to detect isnads. It might also be worthwhile to look into modifying the retrieval model used in the above experiments to use more generalizable document representations than simple bag of ngram representations, which may generalize better to unseen texts.

[3] This list of common starting terms was created in collaboration with a domain expert in the field of Islamic religious studies. Common beginning terms include حدثنيه, حدثنا, سئل, and قال.

Additionally, once we can reliably identify *isnads* in text at scale, several new research questions about the social networks these chains represent become feasible, since *isnads* can be thought of as textual descriptions of a social network of individuals involved in the dissemination of a given piece of information. A reasonable first step would be to take a single *isnad* and extract the names of the individuals involved and the relationship between them. With that accomplished, we could then attempt to infer the social network in which the information is spread by linking names to the individuals they represent using the evidence provided by multiple *isnads* to differentiate between individuals when, for instance, one name is shared by multiple people or different names are used to refer to the same individual, essentially creating a social map of the written tradition's evolution over time. To this end, it will be useful to leverage information from biographical dictionaries and other sources of data about the relationships between individuals involved in transmitting the knowledge present in the corpus.

Acknowledgements

This research was completed with funding from firstly the Qatar National Library, Digital Sira Project, and also from the European Research Council under the European Union's Horizon 2020 research and innovation programme (Grant agreement no. 772989, KITAB). Additionally, the authors would like to thank Sarah Savant, Abdul Rahman Azzam, Hamid Reza Hakimi, Kevin Jaques, Lorenz Nigst, Mathew Barber, and Simon Loynes for providing their expertise and time for data annotation, as well as the anonymous reviewers for their insightful comments.

Reference

Abū Dāwūd Sulaymān b. Dāwūd al-Ṭayālisī (d. 204AH/819CE), *Musnad Abī Dāwūd al-Ṭayālisī*, Muḥammad b. ʿAbd al-Muḥasin al-Turkī ed., (Cairo: Dār al-Hijr, 1999), vol. 2, 223-4.

Altammami, Shatha, Atwell, Eric, and Alsalka, Ammar. 2019. Text Segmentation Using N-grams to Annotate Hadith Corpus. In *Proceedings of the 3rd Workshop on Arabic Corpus Linguistics*, 31-39. ACL.

Harrag, Fouzi. 2014. Text mining approach for knowledge extraction in Sahıh al-Bukhari. *Computers in Human Behavior*, 30:558–566.

Foley, John. 2018. Poetry: Identification, Entity Recognition, and Retrieval. Doctoral dissertation. University of Massachusetts Amherst, Amherst, USA.

Graves, Alex and Schmidhuber, Jurgen. 2005. "Framewise phoneme classification with bidirectional LSTM networks." In *Proc. IJCNN*.

Hochreiter, Sepp and Schmidhuber, Jürgen 1997. "Long short-term memory". *Neural Computation*. 9 (8): 1735–1780. doi:10.1162/neco.1997.9.8.1735.

Lafferty, John, A. McCallum, and F. Pereira. 2001. "Conditional Random Field: Probabilistic Models for Segmenting and Labeling Sequence Data." In *ICML 18*.

Lample, Guillaume, Ballesteros, Miguel, Subramanian, Sandeep, Kawakami, Kazuya, and Dyer, Chris. 2016. "Neural Architectures for Named Entity Recognition." In *Proceedings of NAACL 2016*.

Lorang, Elizabeth, Soh, Leen-Kiat, Datla, Maanas Varma, and Kulwicki, Spencer. 2015. "Developing an Image-Based Classifier for Detecting Poetic Content in Historic Newspaper Collections." In *The Magazine of Digital Library Research* 15(7). DOI: 10.1045/july2015-lorang.

Maraoui, Hajer, Haddar, Kais, and Romary, Laurent. 2018. "Segmentation tool for hadith corpus to generate TEI encoding." In *International Conference on Advanced Intelligent Systems and Informatics*, 252–260. Springer.

Pennington, Jeffrey, Socher, Richard and Manning, Christopher D. 2014. "GloVe: Global Vectors for Word Representation." In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Ritter, Alan, Clark, Sam, Mausam and Etzioni, Oren. 2011. "Named Entity Recognition in Tweets: An Experimental Study."

Romanov, Maxim & Seydi, Masoumeh. (2019). OpenITI: A Machine-Readable Corpus of Islamicate Texts (Version 2019.1.1) [Data set].

Siddiqui, Muazzam, Saleh, Mostafa, and Bagais, Ahmed. 2014. Extraction and visualization of the chain of narrators from hadiths using named entity recognition and classification. *Int. J. Comput. Linguist. Res,* 5(1):14–25.

Tjong Kim Sang, Erik F. and De Meulder, Fien. (2003) "Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition." In *Proceedings of CoNLL-2003*.