Extraction and Visualization of the Chain of Narrators from Hadiths using Named Entity Recognition and Classification

Article · April 2014

Muazzam Ahmed Siddiqui
King Abdulaziz University

Mostafa Saleh
King Abdulaziz University

Extraction and Visualization of the Chain of Narrators from Hadiths using Named Entity Recognition and Classification

Muazzam Ahmed Siddiqui, Mostafa El-Sayed Saleh, Abobakr Ahmed Bagais
King Abdulaziz University
Saudi Arabia
{maasiddiqui, msherbini}@kau.edu.sa, abobakr.a.2012@gmail.com

ABSTRACT: *A Hadith is a report of the deeds or sayings of the prophet Muhammad. Each of these reports were orally transmitted from one person to another till it reached a person who recorded the report along with the chain of transmission. We present a system to automatically extract the chain of narrators from a Hadith through Named Entity Recognition and Classification, and present these transmission chains as a network. In a Hadith, the name of a person may appear as a narrator or as someone who is mentioned in the Hadith. This distinction of names is important as identifying and evaluating the narrators is an important part of Hadith studies. We manually annotated a large Hadith corpus with named entities and used a set of keywords and special verbs to train machine learning algorithms for named entity recognition and classification. The keywords and special verbs identified the context surrounding the tokens labeled as named entities. We compared the performance of different classifiers including generative (Naïve Bayes), and discriminative (K-nearest neighbour and decision tree) and were able to achieve a 90% precision and 82% recall for the named entities. The classifiers were evaluated on a different corpus within the same domain that resulted in an 80% precision and 73% recall. The best classifier was used to label a bigger Hadith corpus and the narrators names thus identified from each Hadith were concatenated to create a chain of narration from the Hadith. These chains were represented as a graph of narrators in the end.*

Keywords: Named Entity Recognition, Arabic Natural Language Processing, Machine Learning, Hadith Text Mining, Network Visualization, Graph Mining

Received: 1 November 2013, Revised 13 December 2013, Accepted 20 December 2013

1. Introduction

Named Entity Recognition (NER) is defined as the recognition of named entities such as people, places, organizations etc. from an unstructured text (Gaizauskas & Wilks, 1997). The term "*Named Entity*" was first introduced in 1995 by the Message Understanding Conference (MUC-6), (Grishman & Sundheim, 1996) under the Information Extraction (IE) paradigm. Identifying information units was realized to be an essential component of IE. These information units include names of persons, organizations, and locations, and numeric expressions such as time, date, money, and percentages. (Nadeau & Sekine, 2007). Before 1996 significant research was conducted by Lisa F. Rau (Rau, 1991) to extract proper names from texts. The work is often cited as the one of the earliest examples of the NER.

NER finds its application in NLP and related areas including information retrieval (Thompson & Dozier, 1997), machine translation (Babych & Hartley, 2003), question answering (Ferrández, Ferrández, Ferrández, & Muñoz, 2007) and text clustering (Toda & Katoaka, 2005). Arabic NLP, in general, and Arabic NER, in specific, are relatively new comers to the field (Habash, 2010). The task is more challenging in Arabic because of the unavailability or absence of preprocessing tools (Mustafa, Abdalla, & Suleman, 2008) and the inflectional nature of the language itself (Babych & Hartley, 2003). With respect to NER (Benajiba, Diab, & Rosso, 2008) described three major obstacles in dealing with Arabic. These include absence of short vowels (vocalization), absence of capital letters in orthography and sparseness, the latter being a direct consequence of Arabic being a morphologically rich language. The absence of capital letters is exemplified in Table 1.

where the words بريد (transliteration: bryd, meaning: mail) and بولندا (transliteration: bwlndA, meaning: Poland), both start from the same letter ب (transliteration: by) but unlike English, there is no capitalization of the letter for the second word, which is the name of a country. The second example in Table 1. contains the words دجاج (transliteration: djAj, meaning: chicken) and دبي (transliteration: dby, meaning: Dubai), and it is clear that the words start from the same letter د (transliteration:), but there is no capitalization for the second word which is the name of a city. Table 2. displays the example of a sentence where the first and second words start from the same letter, but there is no capitalization of the letter in the first word. The absence of short vowels is displayed in Table 3. where the diacritic marks used for short vowels are not used, as it is common in Modern Standard Arabic.

No	Word	Transliteration	Meaning	Type
Example 1	بريد	bryd	mail	Common noun
	بولندا	bwlndA	Poland	Named entity
Example 2	دجاج	djAj	chicken	Common noun
	دبي	dby	Dubai	Named entity

Table 1. Examples Explaining the Absence of Capital Letters in Arabic

Sentence	Meaning
ارسل العميد رسالة فى هذا اليوم السعيد	The dean sent a message at this happy day

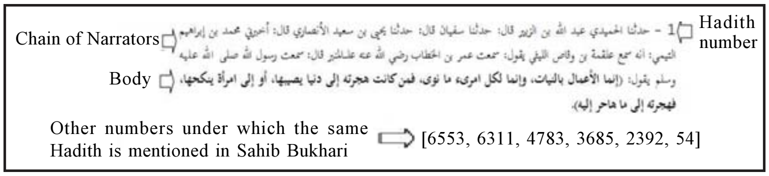
Table 2. Example Showing the Absence of Capital Letters at the Start of a Sentence

No	Word	Transliteration	Meaning	Type
Example 1	صرح	SaroH	edifice	Common noun
	صرح	SaraHa	declared	Past Verb
Example 2	ذهب	*ahabo	gold	Common noun
	ذهب	*ahaba	went	Past verb

Table 3. Example Showing Absence of Short Vowels Leading to Lexical Ambiguity

The term *Hadith* (plural: *Hadiths*) is used to report the saying or an act or tacit approval or criticism ascribed either validly or invalidly to the Islamic prophet Muhammad (peace be upon him) (Islahi, 1989). *Hadiths* are regarded by traditional Islamic schools of jurisprudence as important tools for understanding the Quran and in matters of jurisprudence. These sayings were transmitted by the Prophet’s companions to later generations and were authenticated and recorded in collections along with the names of people (narrators) involved in the transmission process. A recorded *Hadith* consists of two parts, a chain of narration called *sanad* and the actual text of the *Hadith* called *matan*. Figure 1 displays a *Hadith* from Sahih Bukhari with the *sanad* (chain of narrators) and *matan* (body) labeled. The authentication process mainly consists of evaluating the narrators and the study is termed as ilm al-rijal (biographical evaluation; literal: knowledge of men) (Islahi, 1989). Besides narrators, a *Hadith* may contain names of people who were not part of the transmission process. We will refer to the former as Narrator and latter as Person in this paper. This research aims to create a network of narrators by identifying the names of people in *Hadith* collections, tag them as either Narrator or Person, create a chain of narrators from each *Hadith*. Following are the major contributions made by this research.

- Identification of all the forms in which the name of a person may appear in a document (*Hadith*)
- Create a chain of narrators from each *Hadith*
- Create a network of *Hadith* narrators
- Usage of a large corpus with more than 45K tokens
- Identification of intuitive contextual patterns for named entity recognition in a *Hadith*
- Comparison of different machine learning techniques
- Evaluation of classifiers on a new corpus in the same domain


1 - حدثنا الحميدي عبد الله بن الزبير قال: حدثنا سفيان قال: حدثنا يحيى بن سعيد الأنصاري قال: أخبرني محمد بن إبراهيم التيمي: أنه سمع علقمة بن وقاص الليثي يقول: سمعت عمر بن الخطاب رضي الله عنه على المنبر قال: سمعت رسول الله صلى الله عليه وسلم يقول: (إنما الأعمال بالنيات، وإنما لكل امرىء ما نوى، فمن كانت هجرته إلى دنيا يصيبها، أو إلى امرأة ينكحها، فهجرته إلى ما هاجر إليه).

[6553, 6311, 4783, 3685, 2392, 54]
Hadith number
Chain of Narrators
Body
Other numbers under which the same Hadith is mentioned in Sahib Bukhari

Figure 1. A *Hadith* from Sahih Bukhari labeled with its different components

2. Literature Review

One of the first research papers in the field was presented by (Rau, 1991). They built a system to "*extract and recognize* [company] *names*" using heuristics and handcrafted rules. Early work formulated the NER problem as recognizing "*proper names*" in general (Coates-Stephens, 1992), (Thielen, 1995). The term Named Entity (NE), was first introduced in 1995 by the Message Understanding Conference (MUC-6), (Grishman & Sundheim, 1996). Early NER systems mostly relied on hand-crafted rule based algorithms while the current dominant technique is to use machine learning algorithms including supervised, semi-supervised and unsupervised learning methods (Nadeau & Sekine, 2007). In this section, we present a brief review of Arabic NER systems using rule based and machine learning methods and earlier attempts on applying NER to *Hadith* corpora.

Arabic rule-based NER systems mostly relied on two resources; a set of rules and look-up gazetteers. One of the earliest works in Arabic Named Entity Recognition, called TAGARAB, (Maloney & Niv, 1998), was indeed a rule based system that combined a morphological analyzer with a pattern matching module. The morphological analyzer used regular expressions while the pattern matching module used a set of rules to tag the output of morphological analyzer with the named entity category. They reported an average precision of 89.5% and average recall of 80% on four named entity categories. Another early work presented by (Abuleil & Evens, 1998) built a lexicon automatically by tagging Arabic newspaper text. Within the process, they not only identified proper nouns but classified them into different named entity categories too. A rule-based system was designed to identify verb, noun and proper noun phrases and used affixes of the words in the phrase to tag each word with its part of speech. No experimental results were provided. The system was described in more detail in (Abuleil & Evens, 2002). A small corpus of 100 documents was used to evaluate the system and 100% precision and 94% recall for proper nouns was reported. (Abuleil, 2004) further extended the work by representing the identified phrases using directed graphs and using a set of rules to tag named entities. The system was evaluated on a corpus of 500 news articles and achieved a 91% average precision and 78% average recall. A combination of rules and lexicons was used by (Shaalan & Raza, 2009) to build NERA (Named Entity Recognition for Arabic). The lexicons include gazetteers for person, location and organization names. They achieved 91.6% average precision and 93.5% average recall on the 10 named entity categories in their corpus. Another such system was presented by (Al-Shalabi, Kanaan, Al-Sarayreh, Al-Ghonmein, Talhouni, & Al-Azazmeh, 2009) where they extracted proper nouns in Arabic using a set of rules. These rules were based upon a list of keywords and special verbs. The system was evaluated on a small corpus of 20 newspaper articles and achieved 86% precision. (Elsebai, Mezaine, & Belkredim, 2009) developed a rule-based system where they added several lists to GATE to identify person names in Arabic. These lists include location and organization names, special verbs and keywords to identify person names. The system achieved 93% precision and 86% recall in a corpus consisting of 700 news articles.

For Arabic NER using machine learning techniques, (Benajiba, Rosso, & Ruiz, 2007) presented ANERsys, an Arabic NER system based-on n-grams and maximum entropy. They developed location, person and organization gazetteers to improve the accuracy of the system. They were able to achieve an overall 63% precision and 49% recall. The performance of ANERSys was improved by replacing the maximum entropy model with condition random fields and experimenting with different features sets (Benajiba & Rosso, 2008). Another important attempt at using machine learning techniques for NER was by (Benajiba, Diab, &

Rosso, 2009) where they used language independent and language specific features to train a support vector machine classifier. The system was trained and tested on four different corpora with different combinations of feature sets. (Abdallah, Shaalan, & Shoaib, 2012) improved NERA (Shaalan & Raza, 2009) by combining the rule-based system with a decision tree and reported better performance on ANERcorp corpus than (Benajiba & Rosso, 2008).

The application of computational linguistics techniques to Islamic religious text is new. For *Hadith* NER, (Harrag, El-Qawasmeh, & Al-Salman, 2011) used a finite state transducer to extract named entities from prophetic narrations. They used the same original corpus as ours and achieved overall precision and recall of 71% and 39% respectively. A rule based approach was used by (Azmi & Badia, 2010), where they generated grammar rules from *Hadith* corpora and used them for parsing. The system was tested on a small corpus of 90 documents (*Hadith*) and achieved 86.7% success rate.

3. Data

A number of *Hadith* collections have been compiled by different Muslim scholars. A group of six of these collections is termed as *Saha Satta* (The Authentic Six). Our training corpus came from one of the most authentic and widely used collection from the authentic six called Sahih Bukhari (Ibn al-Salah, 2000). Besides Sahih Bukhari, we used another *Hadith* collection called, Musnad Ahmed as a test corpus. In the next subsections, we will explain the corpus, preprocessing and the feature extraction steps.

3.1 Corpus

> The *Hadiths* in Sahih Bukhari are categorized according to non mutually exclusive topics, resulting in the presence of the same *Hadith* under many topics. The total number of *Hadiths* in the edition used is 7124 including duplicate *Hadiths*. The *Hadiths* are numbered in the collection and each *Hadith* is additionally labeled with all the numbers under which it is found in collection, cf.

Figure 1. A *Hadith* from Sahih Bukhari labeled with its different components

In our corpus, each token was tagged with one of the following five classes:

B-NARR: The Beginning of the name of a NARRator

I-NARR: The continuation (Inside) of the name of a NARRator

B-PERS: The Beginning of the name of a PERSon

I-PERS: The continuation (Inside) of the name of a PERSon

O: Not a named entity (Other)

The tagging was done by a native Arabic speaker. We chose to label individual tokens instead of labeling a sequence of tokens as a named entity. In the latter case a preprocessing step is required to mark a sequence of tokens as a noun phrase and, hence, a candidate for named entity. Co-references were not resolved and only the literal name strings were labeled as named entities. There are 3275 instances of the named entity type NARRator and 1259 instances of the type PERSon in the corpus. Table 4. displays the label (class) distribution in the corpus. It is evident that the task is multiclass classification with unbalanced classes.

Class	No of tokens	Proportion
B-NARR	3275	7.1%
I-NARR	3694	8.0%
B-PERS	1259	2.7%
I-PERS	787	1.7%
O	36882	80.4%

Table 4. Class Distribution of Tokens in the Corpus

3.2 Preprocessing

For preprocessing, we only applied normalization to remove any diacritic marks. Stemming was applied to match tokens to the items in the lists provided in table 3. No POS tagging and/or noun phrase extraction was applied for two main reasons. One, because the Arabic text processing tools are designed for MSA (Modern Standard Arabic) and our corpus is in classical Arabic and two, because the available tools are not very accurate. We used the Stanford POS tagger, without any retraining on our corpus and a manual inspection of resulting POS tags revealed a number of errors. To confirm this we computed the entropy of the class distribution for different part of speech tags. For ease of interpretation, we combined the four named entity tags into one NE tag, resulting in a binary classification problem with a maximum entropy value of one for equal class distribution.

Table 5. displays the entropy values for some of the parts of speech tags, indicating that the POS tagging suffered from errors. Had the tagging being correct, we would expect a lower value of entropy for proper nouns, as they are essentially, named entities.

POS Tag	POS	Entropy
NN	common noun	0.598
NNP	proper noun, singular	0.872
PUNC	Punctuation	0.716
VBD	Perfect verb	0.730

Table 5. Entropy for Different POS Tags Indicating Inaccuracies in the Tagging

3.3 Features

In the *Hadiths* collections, a specific format was usually used to report the *Hadith*. We exploited this format to identify candidates for named entity recognition. We defined a feature as an attribute-value pair, which was deemed true, if the attribute took a particular value, false otherwise. More formally, a feature was defined as a Boolean valued function $F(x, y)$, which returned true if x took the value y, false, otherwise. Following is the terminology that we used in defining the features.

n = Token index

C_n = Token at index n

Fd_n = Feature d corresponding to the token at index n

Next, we will define the features that we devised

preceded_by_reporting_verb: The current token was immediately preceded by a reporting verb. This feature implements the relationship given by (1).

$$F1_n = \begin{cases} 1 \; \mathit{if} \; C_{n-1} \in A \; \mathit{and} \; C_n \notin A \\ 0 \quad \text{otherwise} \end{cases} \tag{1}$$

name_continuation: The feature is true if the previous token was preceded by a reporting verb or if the feature was true for the previous token. It is false if the current token is from lists A, C, D or F (Table 5). In the case of an n-word string in the lists, the current token was concatenated with the next n –1 tokens and the longest substring match was sought. This features implements the relationship given by (2).

$$F2_n = \begin{cases} 1 \; \mathit{if} \; F1_{n-1} \text{ or } F2_{n-1} \\ 0 \quad \text{else if } C_n \in (A \text{ or } C \text{ or } D \text{ or } F) \end{cases} \tag{2}$$

part_of_arabic_name: The current or previous token was part of an Arabic name representing nasab (son/daughter of), kunya (father/mother of), or nisbah (family name). This feature implements the relationship given by (3).

$$F3_n = \begin{cases} 1 \; \mathit{if} \; (C_n \text{ or } C_{n-1}) \in B \\ \text{or } ((C_{n-2} \in B) \\ \text{and substr } (C_n, 0, 2) = \text{“ال”}) \\ 0 \quad \text{otherwise} \end{cases} \tag{3}$$

succeeded_by_companion_honorific: The current token was succeeded by the honorific reserved for the companions of the prophet. This feature implements the relationship given by (4).

$$F4_n = \begin{cases} 1\ if\ C_{n+1} \in D \\ 0 \ \ \text{otherwise} \end{cases} \tag{4}$$

succeeded_by_prophet_honorific: The current token was succeeded by the honorific reserved for the prophet. This feature implements the relationship given by (5).

$$F5_n = \begin{cases} 1\ if\ C_n \in E \\ 0 \ \ \text{otherwise} \end{cases} \tag{5}$$

after_the_prophet: A flag indicating that a word form list E, Table 6. has been identified. This feature was used to distinguish between a narrator (B-NARR or I-NARR) and a person (B-PERS or I-PERS). Usually a *sanad* (chain of narration in a *Hadith*) ends at the Prophet. Any name mentioned after the Prophet is a likely candidate for a person (B-PERS or I-PERS) in our corpus. This feature implements the relationship given by (6).

$$F6_n = \begin{cases} 1\ if\ C_m \in E\ such\ that\ m < n \\ 0 \ \ \text{otherwise} \end{cases} \tag{6}$$

preceded_by_arabic_greeting: The current token was preceded by the Arabic greeting word. This feature implements the relationship given by (7).

$$F7_n = \begin{cases} 1\ if\ C_{n-1} \in G \\ 0 \ \ \text{otherwise} \end{cases} \tag{7}$$

mention_of_prophet: The current and previous tokens combination refers to the Prophet as in "*Messenger of Allah*" or "*Prophet of Allah*".

$$F8_n = \begin{cases} 1\ if\ C_n = \text{الله} \text{ and } (C_{n-1} \in \text{نبي}\ or\ C_{n-1} \in \text{رسول}) \\ 0 \ \ \text{otherwise} \end{cases} \tag{8}$$

prefix_of_arabic_name: The current or previous token is or contain the most common prefix of Arabic names.

$$F9_n = \begin{cases} 1\ if\ C_n \text{ or } C_{n-1} = \text{“عبد”} \\ 0 \ \ \text{otherwise} \end{cases} \tag{9}$$

full_string: the token itself

4. Named Entity Recognition

We trained three different classifiers including Naive Bayes (NB), Decision Tree (DT) and K-Nearest Neighbour (KNN) for the NERC task. We did not opt for the one-vs-all or one-vs-one strategies of handling multiple classes, where an *n*-ary classification problem is decomposed into n binary classification problems. The input to the classifiers was of the form (*F*, *C*), where *F* was a feature vector consisting of the 7 features described in the previous section and *C* was the class label. For evaluation, we used an n-fold cross validation method. This method splits the input data *D* into *n* mutually exclusive subsets or "*folds*", D_1, D_2, …, D_n. Training and testing is performed *n* times. In iteration *i*, D_i is used for testing and the rest of the partitions, collectively, are used for training. The final accuracy measure is the average of n iterations. For our experiment we used $n = 10$. It should be noted that the subsets were created at the *Hadith* level and not at the individual token level to retain the context.

We used the MUC scoring to evaluate the performance of our system. In MUC evaluation, an NER is scored on two axes: its ability to find the correct entity type (TYPE) and its ability to find the exact text boundaries (TEXT). A TYPE is considered correct if the entity is assigned the correct category, irrespective of the boundaries as long as there is an overlap. On the other hand, a TEXT is considered correct if the boundaries match exactly irrespective of the category of the entity. In our corpus, two TYPEs were present, NARRator and PERSon. Results are reported using precision, recall and f_1 - measure, given by equations (10), (11) and (12). For overall precision, the number of correct answers includes both correct TYPE and TEXT and the number of predicted entities includes both predicted TYPEs and predicted TEXTs.

List	Type	Arabic	Transliteration	Meaning
A	Reporting verbs	ذكر قال سمع عن قول	*kr qAl smE En qwl	Said Said Hearing About Say
B	Part of Arabic name	ابن/بن بنت اب ام	Bn/ Abn Bnt Ab Um	Son of Daughter of Father of Mother of
C		Punctuation		
D	Companion honorific	رضي الله عنه رضي الله عنها رضي الله عنها	rDy Allh Enh rDy Allh EnhA rDy Allh Enhm	May Allah be pleased with him May Allah be pleased with her May Allah be pleased with them
E	The Prophet	رسول نبي أبا القاسم/ أبو القاسم	rswl nby >bA AlqAsm/ >bw AlqAsm	Messenger Prophet Father of Qasim (Teknonym of the Prophet)
F	The Prophet honorific	صلى الله عليه وسلم	SlY Allh Elyh wslm	Peace be upon him
G	Arabic greeting	يا	yA	O (as in O people)

Table 6. Lists Used in Feature Extraction

$$Precision = \frac{No\ of\ correct\ answers}{No\ of\ predicted\ entities} \tag{10}$$

$$Recall = \frac{No\ of\ correct\ answers}{No\ of\ actual\ entities} \tag{11}$$

$$F_1 - \text{measure} = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{12}$$

Table 7. displays the results for the precision, recall and f1-measure for each classifier. Table 8. breaks these evaluation measures for TYPE and TEXT. The MUC scoring does not report the accuracy of each TYPE separately. We computed the precision, recall and f1-measure of each named entity type. Table 9. displays the results for the NARRator and the PERSon classes separately.

To find out if these differences in the results displayed inTABLE VII. are statistically significant, we compared the classifiers in a pairwise fashion using paired t-test. We computed the t-statistic with 9 degrees of freedom and 5% significance level for the 10-fold cross validation method used. TABLE X. displays the better classifier with a 5% margin of error for each pairwise comparison. The difference between decision tree and k-nearest neighbor for f1-measure was not statistically significant to declare a winner.

Classifier	Prec	Recl	F_1
NB	0.72	**0.90**	0.80
DT	**0.90**	0.82	0.86
KNN	0.83	0.88	**0.85**

Table 7. Overall Precision, Recall And F1-measure

	TYPE			TEXT		
Classifier	Prec	Recl	F_1	Prec	Recl	F_1
NB	0.75	**0.93**	0.83	0.70	0.87	0.77
DT	**0.92**	0.83	0.87	**0.89**	0.80	**0.84**
KNN	0.85	0.90	**0.88**	0.81	**0.95**	0.83

Table 8. Precision, Recall and F_1- measure of Type and Text

	NARRator			PERSon		
Classifier	Prec	Recl	F_1	Prec	Recl	F_1
NB	0.77	**0.98**	0.86	0.69	**0.81**	0.74
DT	**0.91**	0.90	0.91	**0.94**	0.66	0.77
KNN	0.88	0.94	**0.91**	0.77	0.79	**0.78**

Table 9. Precision, Recall and F_1- measure of Each Named Entity Type

Compared classifiers	Best classifier for Prec	Best classifier for Recl	Best classifier for F1
NB vs DT	DT	NB	DT
NB vs. KNN	KNN	NB	KNN
DT vs. KNN	DT	KNN	None

Table 10. Classifier Comparison Results for Precision, Recall and F_1- measure

Among the classifiers naïve Bayes suffered from a low precision but gave highest recall rates. The discriminate classifiers (decision tree and k-nearest neighbor) performed better with higher f1-mesure, although the recall was usually lower than that of naïve Bayes. The low precision indicates a high false positive while the high recall indicates a low false negative rate for the Naïve Bayes. The classifier had high tolerance for positive cases, and a number of O (Other) type tokens were marked incorrectly as belonging to a named entity. The results can be compared to (Harrag, El-Qawasmeh, & Al-Salman, 2011) and (Azmi & Badia, 2010), where NER were built to extract narrator names from *Hadith* collections. We used a bigger corpus and were able to achieve higher precision and recall rates than (Harrag, El-Qawasmeh, & Al-Salman, 2011) and (Azmi & Badia, 2010). In addition, we identified all the different ways a name of a person may appear in a *Hadith*.

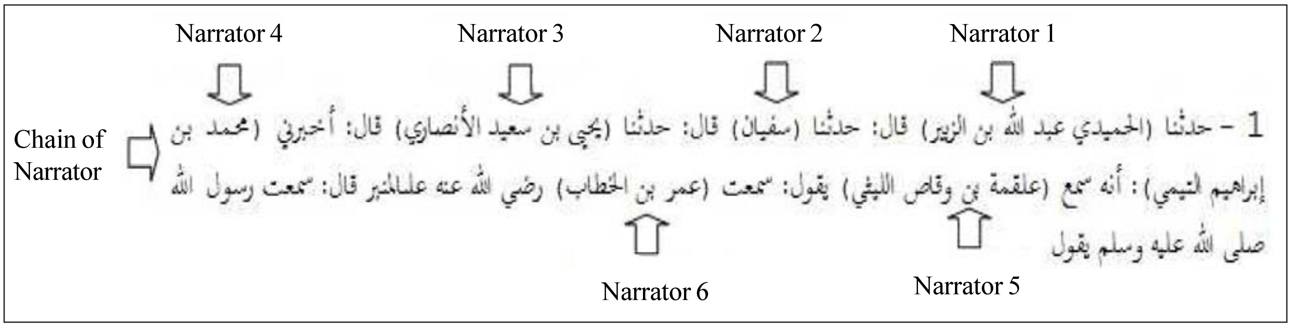
To test the accuracy of the NERC system on a different corpus in the same domain, we selected the classifier with the highest accuracy on the Sahih Bukhari corpus and used it to label a new corpus, which was not part of the training process. The new corpus came from the *Hadith* collection, called Musnad Ahmed and a small subset of it containing about 5K tokens was manually labeled for evaluation. Table 11. displays the results.

5. Extraction of Narrator Chain and Visualization

To extract the narrators chain, we selected the classifier with highest precision and used it to label the entire Sahih Bukhari corpus. A single narrator is extracted from a *Hadith* by identifying a sequence of labels with a starting B-NARR tag followed by zero or more I-NARR tags. Figure 2 displays the chain of narrators from a *Hadith* and it is evident that, once identified, the individual narrators can be concatenated to construct the chain. Figure 3 displays the chain extracted from the *Hadith* mentioned in Figure 3.

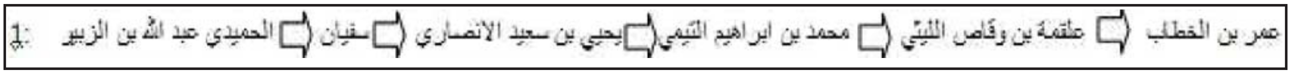
Criteria	Prec	Recl	F_1
TYPE	0.74	0.84	0.79
TEXT	0.59	0.66	0.62
NARRator	0.83	0.98	0.90
PERSon	0.51	0.52	0.51
Overall	0.66	0.75	0.71

Table 11. Precision, Recall and F_1- measure for the Test Corpus



1 – حدثنا (الحميدي عبد الله بن الزبير) قال: حدثنا (سفيان) قال: حدثنا (يحيي بن سعيد الأنصاري) قال: أخبرني (محمد بن
إبراهيم التيمي): أنه سمع (علقمة بن وقاص الليثي) يقول: سمعت (عمر بن الخطاب) رضي الله عنه علىالمنبر قال: سمعت رسول الله
صلى الله عليه وسلم يقول

Figure 2. Chain of narrators from a *Hadith*. Brackets were introduced to mark the boundaries of named entities



عمر بن الخطاب ⇦ علقمة بن وقاص الليثي ⇦ محمد بن ابراهيم التيمي ⇦ يحيي بن سعيد الانصاري ⇦ سفيان ⇦ الحميدي عبد الله بن الزبير :1

Figure 3. Chain of narrators extracted from the *Hadith* mentioned in Figure 2

We applied entity resolution to the extracted narrator chains as a post processing step to consolidate the different mentions of the same person. We identified four specific problems in this regards

1. Different *spelling* of the name of the person, e.g. شعبة vs شعبه.

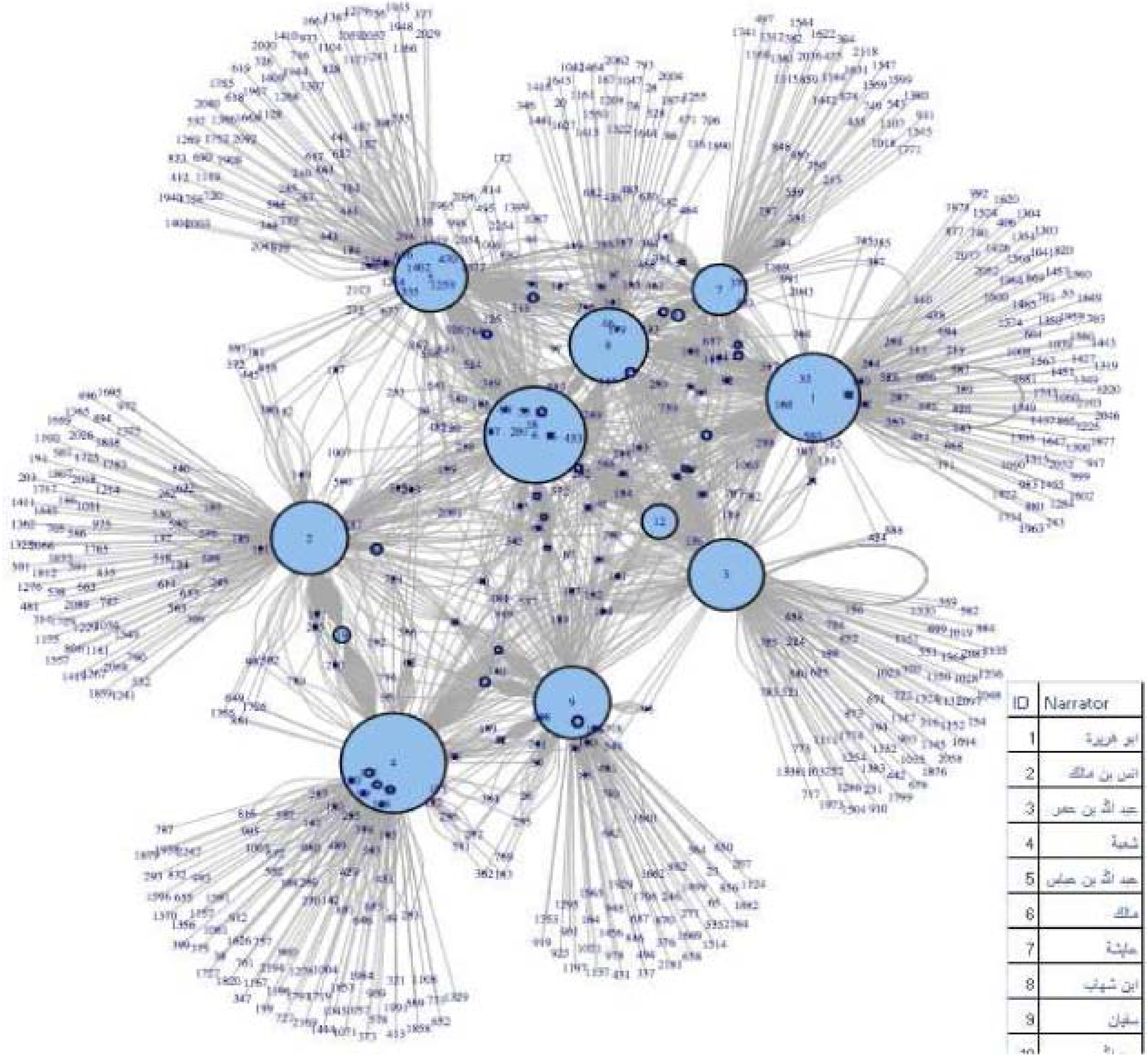
2. Different versions of *kunya* used to address the same person, e.g. عبد الله ابن عمر and ابي هريرة refers to the same narrator. Similarly عبد الله ابن عمر and عبد الله ابن عمر refers to the same narrator.

3. Mention of the full name of the person vs partial name that is still identifiable, e.g. ابن عمر and عمر عبد الله ابن refers to the same narrator.

4. The use of the terms, ابيه (his father) and ابي (my father) in the chain, where the narrator quoted from his father without mentioning the latter's name

To solve the first problem, we applied letter normalization. For the second problem, we identified all the different versions of a *kunya* and replaced them with one single instance. To identify the different versions of *kunya* in narrator names, we computed Levenshtein distance (Levenshtein, 1966) between each pair of names and manually inspected the names that had an edit distance of one. For the third problem, a manual inspection was performed to identify full vs partial names. We identify the fourth problem as a pronoun resolution problem and resolved it by concatenating the term with the name of the narrator immediately preceding it.

For visualization, the chains of narrators were converted to a graph, with nodes representing narrators and edges representing the transmission link between two narrators. We used the *igraph* (Csardi & Nepusz, 2006) package in *R* (Team, 2013) to create the visualization. The chain of narrators were converted to the edge format, e.g. the chain A->B->C was decomposed to two edges A->B and B->C. Figure 4 displays the network of narrators for the ten most prolific narrators from Sahih Bukhari. The size of the vertex represents the number of *Hadiths* narrated by the narrator. The scarcity of the space forced us to label the vertices with the IDs of the narrators, instead of their full names. At the bottom right corner of the picture, a legend is provided to link the IDs with the names of the top ten narrators only.



ID	Narrator
1	أبو هريرة
2	أنس بن مالك
3	عبد الله بن عمر
4	شعبة
5	عبد الله بن عباس
6	مالك
7	عائشة
8	ابن شهاب
9	سفيان
10	

Figure 4. Network of narrator chain showing all the links for the top 10 narrators

6. Conclusion and Future Work

This paper presented a system to create a network of *Hadith* narrators by automatically extracting the sequence of narrators from *Hadith* and converting these sequences to the graph format. From each *Hadith*, the narrators were extracted through named entity recognition and classification and the extracted named entities were concatenated to form a sequence. For the NERC task, we manually identified a number of contextual rules and converted them to features that were used to train machine learning classifiers. The extracted sequences were converted to graph format where vertices represented narrators and edges represented the transmission link between narrators. Creating the network of narrators opens the gate for deeper analysis by modeling the network as a graph. A number of important network characteristics can be identified through classification and clustering of graph that can give further insight into the narrator network. Chief among them is the community detection, that is, to identify dense interconnected regions within the network representing small groups of narrators involved in transmitting a large number of *Hadiths*. Others include outlier detection that would identify narrator chains isolated from the rest of the community and hub identification that would identify narrators who have the large number of *Hadiths* transmitted through them.

7. Acknowledgements

This project was funded by the Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah under grant no. (126/611/1431). The authors, therefore, acknowledge with thanks DSR technical and financial support.

References

[1] Abdallah, S., Shaalan, K., Shoaib, M. (2012). Integrating Rule-Based System with Classification for Arabic Named Entity Recognition. *In*: A. Gelbukh (Ed.), CICLing'12 Proceedings of the 13th International Conference on Computational Linguistics and Intelligent Text Processing (p. 311-322). Springer Berlin Heidelberg.

[2] Abuleil, S. (2004). Extracting names from Arabic text for question-answering systems. *In*: Proceedings of the 7th International Conference on Coupling Approaches, Coupling Media, and Coupling Languages for Information Retrieval, (p. 638–647). University of Avignon (Vaucluse), France.

[3] Abuleil, S., Evens, M. (1998). Discovering Lexical Information by Tagging Arabic Newspaper Text. Semitic 98 Proceedings of the Workshop on Computational Approaches to Semitic Languages.

[4] Abuleil, S., Evens, M. (2002). Extracting an Arabic Lexicon from Arabic Newspaper Text. *Computers and the Humanities ,* 36 (2) 191-221.

[5] Al-Shalabi, R., Kanaan, G., Al-Sarayreh, B., Al-Ghonmein, A., Talhouni, H., Al-Azazmeh, S. (2009). Proper Noun Extracting Algorithm for Arabic language. International Conference on IT to Celebrate S. Charmonman's 72nd Birthday.

[6] Azmi, A., Badia, N. (2010). iTree – Automating the Construction of the Narration Tree of Hadiths (Prophetic Traditions). International Conference on Natural Language Processing and Knowledge Engineering (NLP-KE).

[7] Babych, B., Hartley, A. (2003). Improving machine translation quality with automatic named entity recognition. *In*: Proceedings of the 7th International EAMT workshop on MT and other Language Technology Tools, Improving MT through other Language Technology Tools: Resources and Tools for Building MT .

[8] Benajiba, Y., Rosso, P. (2008). Arabic Named Entity Recognition using Conditional Random Fields. *In*: Proc. Workshop on HLT & NLP within the Arabic world. Arabic Language and local languages processing: Status Updates and Prospects, 6th Int. Conf. on Language Resources and Evaluation. Marrakech, Morocco.

[9] Benajiba, Y., Diab, M., Rosso, P. (2008). Arabic Named Entity Recognition using Optimized Feature Sets. *In*: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing.

[10] Benajiba, Y., Diab, M., Rosso, P. (2009). Using Language Independent and Language Specific Features to Enhance Arabic Named Entity Recognition. *The International Arab Journal of Information Technology*, 6 (5) 464-472.

[11] Benajiba, Y., Rosso, P., Ruiz, J. (2007). ANERsys: An Arabic Named Entity Recognition System Based on Maximum Entropy. *Computational Linguistics and Intelligent Text Processing*, (p. 143–153).

Coates-Stephens, S. (1992). The Analysis and Acquisition of Proper Names for the Understanding of Free Text. *Computers and the Humanities,* 26, 441-456.

[12] Csardi, G., Nepusz, T. (2006, 0205). The igraph software package for complex network research. *InterJournal, Complex Systems*, 1695.

[13] Elsebai, A., Mezaine, F., Belkredim, F. (2009). A Rule Based Persons Names Arabic Extraction System. Communications of the IBIMA*,* 11.

[14] Ferrández, S., Ferrández, O., Ferrández, A., Muñoz, R. (2007). The Importance of Named Entities in Cross-Lingual Question Answering. Int. Conf. Recent Advances in Natural Language Processing*, RANLP.*

[15] Gaizauskas, R., Wilks, Y. (1997). Information Extraction: Beyond Document Retrieval. *Memoranda in Computer and Cognitive Science*, 54, 70–105.

[16] Grishman, R., Sundheim, B. (1996). Message Understanding Conference - 6: A Brief History. *In*: Proc. International Conference on Computational Linguistics.

[17] Habash, N. (2010). *Introduction to Arabic Natural Language Processing* (1 ed.). (G. Hirst, Ed.) Morgan and Claypool Publishers.

[18] Harrag, F., El-Qawasmeh, E., Al-Salman, A. (2011). Extracting Named Entities from Prophetic Narration Texts (Hadith). *In*: Proceedings of ICSECS (2), (p. 289–297).

[19] Ibn al-Salah, A. (2000). *Muqaddimah Ibn al-Salah.* Dar al-Ma'aarif.

[20] Islahi, A. (1989). *Mabadi Tadabbur-i-Hadith.* Lahore: Al-Mawrid.

[21] Levenshtein, V. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady ,* 10 (8).

[22] Maloney, J., Niv, M. (1998). TAGARAB:A fast, accurate arabic name recogniser using high precision morphological analysis. *In*: Proceedings of the Workshop on Computational Approaches to Semitic Languages Montreal, (p. 8-15).

[23] Mustafa, M., Abdalla, H., Suleman, H. (2008). Current Approaches in Arabic IR: A Survey. *In*: Proceedings of the 11th International Conference on Asian Digital Libraries: Universal and Ubiquitous Access to Information (ICADL 08) (p. 406-407). Berlin, Heidelberg: Springer-Verlag.

[24] Nadeau, D., Sekine, S. (2007). A survey of named entity recognition and classification. *Linguisticae Investigationes,* 30, 3-26.

[25] Rau, L. F. (1991). Extracting Company Names from Text. *In*: Proceedings of Seventh IEEE Conference onArtificial Intelligence Applications (p. 29–32). IEEE.

[26] Shaalan, K., Raza, H. (2009). NERA: Named Entity Recognition for Arabic. *Journal of the American Society for Information Science and Technology ,* 60 (8) 1652–1663.

[27] Team, R. C. (2013). R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing.

[28] Thielen, C. (1995). An Approach to Proper Name Tagging for German. *In*: Proc. Conference of European Chapter of the Association for Computational Linguistics. *SIGDAT.*

[29] Thompson, P., Dozier, C. (1997). Name Searching and Information Retrieval. *In*: Proc. of Second Conference on Empirical Methods in Natural Language Processing, (p. 134–140).

[30] Toda, H., Katoaka, R. (2005). A search result clustering method using informatively named entities. *In*: Proceeding of the 7th ACM International Workshop on Web Information and Data Management (WIDM).