Arabic Cross-Document NLP for the Hadith and Biography Literature *

Fadi Zaraket and **Jad Makhlouta**
American University of Beirut
{*fz11,jem04*}*@aub.edu.lb*

Abstract

Recently cross-document integration and reconciliation of extracted information became of interest to researchers in Arabic natural language processing. Given a set of documents A, we use Arabic morphological analysis, finite state machines, and graph transformations to extract named entities N_a and relations R_a expressed as edges in a graph $G = \langle N_a, R_a \rangle$. We use the same techniques to extract entities N_b and relations R_b from a separate set of documents B. We use G to disambiguate N_b and R_b and we integrate the resulting entities back into G by annotating the nodes and edges in G with elements from N_b. We apply our approach in an iterative manner. Our results show a significant increase in accuracy from 41% to 93% after applying this cross-document NLP methodology to hadith and biography documents.

1 Introduction

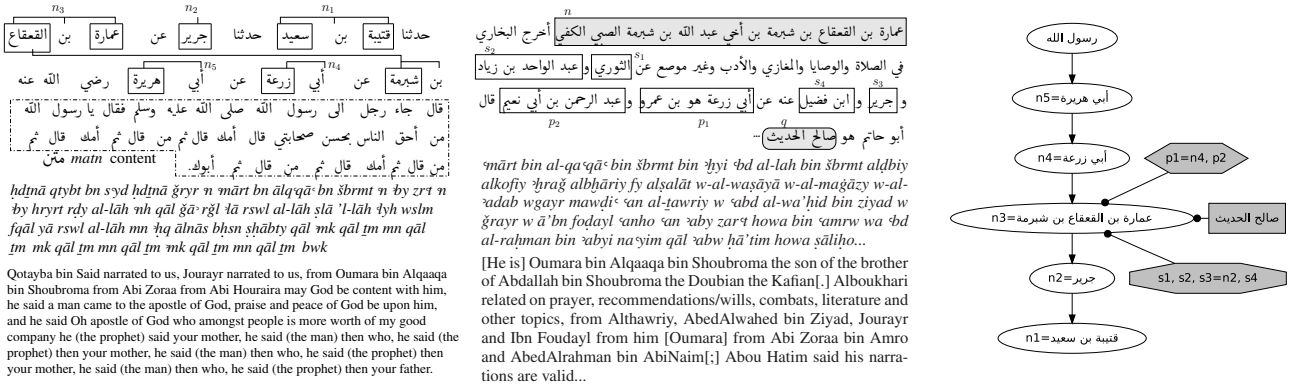
Recently, the *natural language processing* (NLP) community became interested in *cross-document* reconciliation and integration of automatically extracted entities and relations from text. Cross-document NLP studies the problem of coreference of real world entities in separate documents and the use of the detected cross-references to disambiguate, augment, and integrate the extracted entities and relations (Stephanie Strassel and Maeda 2008). *Cross-document structure theory* assumes a set of rhetorical relations amongst paragraphs across documents that are related by topic. The existence of such information helps the tasks of multi-document summarization and information retrieval. For example, unlabeled data helped improve the accuracy of binary classifiers that determine whether a relation exists between a pair of sentences from two separate documents, as well as full classifiers that determine the nature of the relation amongst a taxonomy of 18 relations (Zhang, Otterbacher, and Radev 2003). Work that extracts named entities and relate them over a timeline across Wikipedia documents presented acceptable accuracy results (Bhole et al. 2007). HistoryViz provides a graph based interface to visualize entities and relations extracted from Wikipedia over a timeline of extracted temporal entities (Sipos et al. 2009). Other work used and evaluated an ontology of Wikipedia against the task of cross-document entity coreference (Finin et al. 2009).

In this work, we consider information extraction from sets or related documents where one set complements and facilitates the information extraction task in the other. For example, consider extracting person and location names from security reports investigating incidents and interrogative reports with suspects, and consider checking the consistency of the reports. Also consider documents with travel itineraries and documents with descriptions of travel destinations. Formally, given a large text h, our target is to segment h into several documents $D = \{d_1, d_2, \ldots, d_n\}$, extract entities $E_i = \{e_i^1, e_i^2, \ldots, e_i^m\}$ from d_i, and extract a relation $R \subset E \times E \times L$, where $E = \cup_{i=1,\ldots,n} E_i$ and L is a set of labels. We consider the same problem across two sets of large texts $H = \{h_1, h_2, \ldots, h_k\}$ and $B = \{b_1, b_2, \ldots, b_p\}$ where all texts in H share structural relations that characterize the documents therein and all texts in B share structural relations that characterize documents therein, and documents in H and B are related. In this work, we present techniques based on Arabic morphological features, finite state machines, graph transformations, to solve the problem and we show that using cross-document NLP significantly improves accuracy of entity and relation extraction. We evaluate our method on hadith (H) and biography (B) documents from the Arabic literature where digitized documents are accessible.

Figure 1(a) shows an example حديث *ḥadyṯ*—hadith (we use the ZDMG transliteration style), i.e. a narration related to the prophet Mohammad, with its transliteration and translation, where proper names are in boxes and are connected to form complex names of narrators. A hadith document is structured into a سند *sanad* and a متن *matn*. The *sanad* is a sequence of narrators referencing each other such as $\langle n_1, n_2, n_3, n_4, n_5 \rangle$ in Figure 1(a). قتيبة is the first name of n_1, and سعيد is the name the father of n_1 as the connector word بن (child of) indicates. The *matn* is the actual content of the hadith and is denoted by dotted lines.

The text in Figure 1(b) is an excerpt from the biography of the hadith narrator عمارة بن القعقاع *ʿmārt bin al-qaʿqāʿ* denoted as n. The biography contains information such as

*This work is supported by grant 11-303-0522236 from the Lebanese National Council for Scientific Research (LNCRS).



(a) Hadith (b) Biography (c) Hadith × Biography

Figure 1: Hadith, biography, and narrator graph annotation example.

the birth place, the lifespan, and the credibility of the narrator. It also contains references to his or her professors and students, e.g. p_1 and p_2, and s_1, s_2, s_3 and s_4 from Figure 1(b), respectively. Notice that s_3 is n_2 from Figure 1(a), p_1 is n_4, and n is n_3.

Our contributions. We considered books of unsegmented hadith text to be H, and books of unsegmented narrator biographies to be B. For each hadith book $h \in H$, we computed morphological features and part of speech (POS) tags. We considered the structure of a hadith as a rhetorical relation and encoded it using finite state machines that take the computed features from h and produce segmented hadith documents each with an internal structure similar to that in Figure 1(a). xWe used the same technique for books in B. **(1)** We obtained excellent results for H and less accurate results for B. We considered the computed structure of each hadith, and **(2)** used a novel hierarchical Arabic name distance metric to identify identical narrators, and **(3)** aggregated the structures extracted from H by merging identical narrators to build a narrator graph G. **(4)** We used G to boost entity and relation extraction in B using graph algorithms and obtained significantly better results. In this paper we stress the cross-document aspect of our work where we use cross-document NLP to improve entity and relation extraction from hadith and biography documents.

2 Background

In this section we review the importance of the information retrieval task form hadith and biography texts, Arabic morphological analysis, and graph transformation techniques.

Importance of hadith authentication. The collections of narrations are the second source of jurisprudence after the قرآن *qorʾān* for all Islamic schools of thought. Due to religious and political reasons, writing the narrations was forbidden until the days of the eighth Umayad Calif, عمر بن عبد العزيز *ʿmr bn ʿbd alʿzyz*(717-720 AC), seventy or so years after the death of the prophet (Askari 1984). Consequently, several inconsistencies were introduced to the literature which necessitated a thorough authentication study of a narration before its use in jurisprudence. While different Islamic schools of thought differ on how to interpret the content, they almost all agree not to use narrations with a *sanad* that lacks authenticity for jurisprudence. The authenticity of a hadith depends on the credibility of the narrators as reported in separate biography books. Figure 1(c) provides a graphical illustration of the process of annotating narrator n_3 from the hadith in Figure 1(a) with credibility information q extracted from the biography in Figure 1(b).

Hadith authentication is currently manual and error prone due to the huge number of existing narration books. Al-Azami(1991) cites more than eleven books of digitized narration books each of several volumes, and a dozen other biography and secondary authentication books.

Arabic morphological analysis. Arabic morphological analysis is key to the analysis of Arabic text (Shaalan, Magdy, and Fahmy 2010). Current morphological analyzers (Al-Sughaiyer and Al-Kharashi 2004; Buckwalter 2002) consider the internal internal structure of an Arabic word and compose it into several concatenated *morphemes*, i.e. *stems* and *affixes*. An affix can be a *prefix, suffix,* or an *infix*. The word أَحمَده *ʾaḥmadh* may have two valid morphological analyses. The letter أَ *ʾa* may be a prefix and the word means "I praise him", or The letter أَ *ʾa* may also be part of the stem أَحمَد *ʾaḥmad* (a proper noun) and the word means "his Ahmad". The morphological analysis of حدثنا *ḥdṯnā* "narrated to us", the word that starts the hadith in Figure 1(a), returns the stem حدث *ḥdṯ* "narrated", which is also the stem of other words such as حدثني *ḥdṯny* "narrated to me" and حدثهم *ḥdṯhm* "narrated to them". The stem also shares similar POS and meaning gloss tags with words such as قال *qāl* "said" and أَخبَر *ʾaḫbar* "told".

Graph algorithms. We use the following algorithms.

- *Merge*: takes two nodes n_1 and n_2, removes them from the graph and adds a node $m(n_1, n_2)$ that has the edges of n_1 and n_2 except the induced self edges.
- *Split*: takes a merged graph node $m(n_1, \ldots, n_k)$ that was formed by merging several nodes, and splits m into $m_1(n_1, \ldots, n_i)$ and $m_2(n_{i+1}, \ldots, n_k)$ so that a threshold is met on the distance between the nodes in m_1 and m_2.

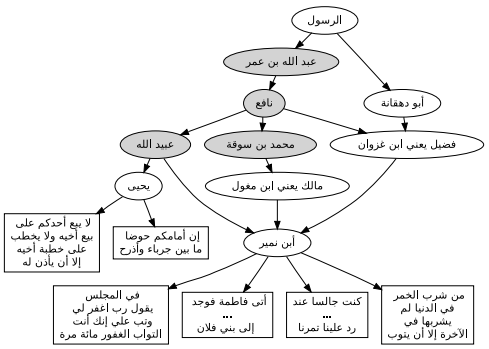


Figure 2: Narrator graph extracted using ANGE.

- *Break cycle*: takes a sequence of nodes that form a cycle $n_1, \ldots, n_k$, identifies one of them whose splitting breaks the cycle and splits it accordingly.
- *k-reachable*: takes two nodes n_1 and n_2 and returns whether n_1 is reachable from n_2 using at most k edges.

3 Annotated narrator graph extractor

In this paper, we present an automatic *annotated narrator graph extractor* (ANGE), from hadith and biography documents using morphology, finite state machines, graph transformations, and cross-document NLP. ANGE first segments a hadith book into narrations, where each narration has a *matn* and a sanad, and each sanad has several narrators. ANGE represents each sanad with a sequence similar to the sequence of ellipse nodes in Figure 1(c). ANGE then takes the extracted sequences and uses a hierarchical name distance metric to identify identical narrators and merges the identical narrators to obtain a narrator graph G. Figure 2 shows a narrator graph extracted using ANGE from text containing six separate hadith documents. The nodes in ellipses are extracted narrator names, and the nodes in boxes are the matn of the hadith documents. ANGE resolves conflicts when generating G using graph cycle breaking algorithms.

ANGE detects narrator names in biography books and orders the detected narrator names according to their appearance in text. For each narrator $n \in G$, ANGE locates the best narrator matches in the narrators extracted from the biography books using the students and professors relation assumption. Those should match the neighbors of n in G. For example, consider n to be the node corresponding to نافع *nāfʿ* in Figure 2. The biography of نافع *nāfʿ* in سير الاعلام والنبلاء *syr ālʿaʿlām wālnblāʾ* (The Biographies of The Figures and the Nobles) contains the names عبيد الله بن يزيد *ʿbyd āllh bn yzyd* , محمد بن سوقة *mḥmd bn swqt* , and عبد الله بن عمر *ʿbd āllh bn ʿmr* that are colored in Figure 2. ANGE annotates n with features extracted from the biography. This process results in segmenting the biography books into their separate biographies. The process can be repeated several times with lower thresholds every time until all biographies with matching narrators in the narrator graph generated from the hadith books are located. The rest of the biographies are not interesting since they do not refer to persons who narrated hadiths.

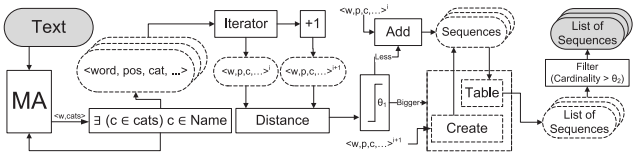


Figure 3: The sequence of narrator extractor diagram.

distance(Narrator n_1, Narrator n_2)

```
List (l1, p1) = canonize(n1)
List (l2, p2) = canonize(n2)
int count=0
for i=0 to min(l1.size, l2.size)
  if (l1[i] == {} XOR l2[i] == {})     continue
  if (¬morphEqual(l1[i], l2[i]))       return ∞
  count++    // l1[i] == l2[i]
foreach poss1 in p1
  foreach poss2 in p2
    if (conflicting(poss1, poss2))     return ∞
    if (reinforcing(poss1, poss2))     count++
return 1 - count/(min(l1.size, l2.size) + min(p1.size, p2.size))
```

3.1 Narrator sequence extraction

The diagram in Figure 3 describes the extraction of narrator sequences from hadith books. ANGE passes the hadith book to an inhouse morphological analyzer and computes a vector $\langle w, cats \rangle$ with morphological features, POS tags, and gloss tags that characterize proper names. ANGE filters the feature vectors and considers only vectors with a feature $c \in Name$ where $Name$ is the set of features that categorize names.

ANGE computes the number of words separating each subsequent pair of names (w_1, w_2). If w_2 is less than θ_1 words distant from w_1, ANGE adds w_2 to the current sequence of names. Otherwise, **(1)** ANGE considers the current sequence that ends with w_1 as a sanad if the number of narrator names in it was $\geq \theta_2$ where θ_2 denotes the minimum number of narrators required for a sanad. **2.** ANGE also creates a new sequence with w_2 as its first element and considers that to be the current sequence.

ANGE also benefits from name connectors that denote family relations such as ابن *ibn*(the child of) which are building blocks in complex narrator names. In addition, it benefits from narration words which act as narrator connectors such as عَن *ʿan* (on behalf of) and حَدَّث *ḥadaṯ* (narrated) and their morphological variations. We manually encoded the narrator sequence extractor of Figure 3 into a finite state transducer.

3.2 The narrator graph extraction

Figure 2 shows a narrator graph that ANGE extracted from six sequences of narrators after merging identical narrators. Two narrator names may refer to the same person but may differ in text and in structure. For example, the names عبد الله بن عمر, ابن عمر, and ابن عمر رضي الله عنه refer to the same person. We use the `distance` morphological and structural metric to compute identical narrators. The (l, p) =`canonize(`n`)` function takes a narrator n and returns the morphological stems of the name components of n ordered with parenthood in the list of names l. Parenthood is inferred from the ابن (son) name connector and its morphological equivalents. `canonize(`n`)` also returns the

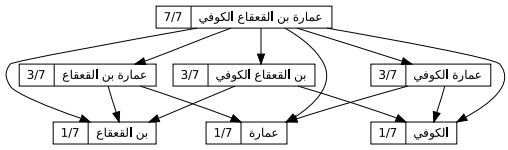


Figure 4: Hash keys and corresponding relevance scores.

qualifying components of n, such as العراقي *ālʿrāqy* (the Iraqi) in the name عبد الله العراقي, in p.

Then `distance` counts the matching name components and the matching credibility qualifying components. It returns ∞ if the morphological features of two name components at the same level do not match. If only one of the name components at the same level is empty such as عبيد الله بن عبد الله بن عمر and عبد الله بن عمر, the metric skips it. For the qualifying components, `distance` uses POS and gloss tags to check whether two qualifiers conflict with each other, such as العراقي *ālʿrāqy* (the Iraqi) and المصري *ālmṣry* (the Egyptian) and returns ∞. It also counts the number of qualifying components that reinforce each other such as العراقي *ālʿrāqy* and الكوفي *ālkwfy* (The Kokian) where Kofā is a city in Iraq. Finally the metric returns the percentage morphological and structural differing components.

The naive brute force method to detect all identical narrators in thousands of sequences extracted from hadith books renders the task of building a complete narrator graph intractable. Instead, ANGE uses a multi-key hash technique that associates every narrator name with the several narrator names that can be built from its components. For example, the narrator name عمارة بن القعقاع الكوفي *ʿomārat bin al-qaʿqāʿ al-kowfiy* is composed of two names, i.e. عمارة and القعقاع, one name connector, i.e. بن, and one possessive qualifier, i.e. الكوفي. Six other structures that refer to the same narrator can be generated out of those components as shown in Figure 4. ANGE considers each of the structures as a hash key valid to match the original narrator and associate a match score with it.

The `buildGraph` Algorithm takes as input a list of narrator sequences C. It iterates over all narrators n and iterates over all the hash keys of n. For each hash key k in n.`keys`, the algorithm looks up the narrator and score pair $matches$ of k in the hash table H_n. For each matching pair (n_h, s_h), the algorithm checks whether n and n_h are of the same generation by checking their corresponding indices in their sequences against a radius parameter r. This refines the search to avoid checking a narrator relating directly to the prophet against a narrator separated from the prophet by at least five narrators. If the pair (n_h, s_h) matches n better than the previously matching pair (n_b, s_b), then n_h is checked for equality against n to protect from hash collisions before it is selected as the best match. Finally, if the score of the best match s_b is bigger than a parameter θ, n and n_h are merged.

```
buildGraph(Narrator C[][])

Hash H_n={}
foreach sequence c in C
  foreach Narrator n in c
    Narrator n_h, n_b
    double s_h, s_b=0          //scores
    foreach k in n.keys()
      NarrScorePair matches[] = H_n.findAndInsert(n,k)
      foreach (n_h,s_h) in matches
        if (|n_h.index-n.index| < r)
          if (s_h × key.score() > s_b)
            if (distance (n, n_h) < ϵ)
              s_b=s_h, n_b=n_h
    if (s_b ≥ θ) mergeNodes(n, n_h)
```

4 Biography detection

ANGE computes l_b, the list of detected narrators in the biography books using similar techniques used in the hadith books, and orders l_b by the order of appearance of the narrators in the biography book. ANGE then considers the rhetorical relation that characterizes the structure of a biography that includes the name subject narrator n of the biography and the narrator names of the professor and student of n. The first approach of ANGE works similarly to Figure 3 and extracts narrator names, professor, and student features, and biographies from the biography text books without the use of the narrator names and the narrator graph extracted from the hadith books. Then, ANGE applies cross-document reconciliation and uses the narrator names and the narrator graph extracted from the hadith books and improves accuracy of entity and relation extraction from biographies significantly.

For each narrator n_i in l_b, $1 \leq i \leq |l_b|$, ANGE uses the multi-key hash technique and locates n_i in the narrator graph. ANGE uses the *k-reachable* graph algorithm and checks whether subsequent narrators,$(\ldots, n_{i-1}, n_i, n_{i+1}, \ldots)$ in l_b are reachable within k steps from each other. If not, then ANGE decides that it crossed a biography boundary. ANGE inspects the cluster of narrators reachable within k steps and selects the center of the cluster n_c as the subject narrator of the biography. The parents and the children of n_c in the narrator graph are the professors and the students of n_c, respectively. The colored nodes in Figure 2 illustrate this process.

Ambiguity happens in case of several matches for n_i in the narrator graph. ANGE resolves the ambiguity using two measures. First ANGE uses a threshold against the hash score to limit the matches. Also, ANGE only considers matches of n_i that are within k steps of the matches of narrator n_{i-1} preceding n_i in l_b.

4.1 Partial narrator graph annotation

In most cases, a scholar is interested in only annotating a partial narrator graph G_p that he or she extracted using ANGE from a selected set of hadith of interest, e.g. the set may contain narrations related in topic. For each narrator n in G_p, ANGE computes $N_s = \{n_s^1, n_s^2, \ldots\}$ and $N_p = \{n_p^1, n_p^2, \ldots\}$ the children and parents of n in G_p. We compute N_b the intersection of $\{n\} \cup N_s \cup N_p$ and l_b, and sorts N_b by the order of appearance in the biography books. ANGE considers the clusters in N_b that contain n and that happen in the same text locality as the candidate biographies of n and ranks them in terms of the number of matches.

5 Related work

Hadith analysis. The iTree (Azmi and Bin Badia 2010) tool extracts narrator sequences from manually segmented narrations. It uses a context free grammar (CFG) to extract narrators and uses the Levenshtein distance metric to compare narrator names. ANGE differs in that it targets the harder problem of segmenting hadith books, does not use a CFG, and instead uses morphological features and FSMs to detect the narrator names, and uses a novel morphological and structural distance metric to check narrator name equality.

Arabic named entity detection. Named entity detection techniques for Arabic exist and use local grammars (Shaalan and Raza 2009; Zaghouani et al. 2010; Traboulsi 2009) along with other boosting techniques to detect named entities. Our cross document integration and reconciliation technique can work directly with the named entities extracted with such techniques.

NERA uses rule-based systems with local grammars (Shaalan and Raza 2009). ANERSys boosts the capabilities of its statistical models to detect named entities by using task specific corpora and gazetteers and ignores POS tags (Benajiba, Rosso, and Benedíruiz 2007; Benajiba, Diab, and Rosso 2008). It also uses morphosyntactic features and a parallel Arabic and English corpora to bootstrap noisy features (Benajiba et al. 2010). The techniques in (Zaghouani et al. 2010; Al-Jumaily et al. 2011; Maloney and Niv 1998) use light morphological stemming for Arabic with local grammars developed for Latin languages, as well as pattern matching. ANGE differs from those techniques in that it computes features based on morphology, POS and gloss tags assigned to all morphemes and (not only to the stems) and passes the feature vectors to manually built finite state machines that detect the target structures. ANGE is not restriced by stop words (Abuleil 2004), or by the expressive power of a local grammar. It does not enumerate all the structures that express the target entities.

Cross-document NLP. Similar to (Zhang, Otterbacher, and Radev 2003) we assume structural relations exist amongst entities in a document. ANGE differs in that it extracts the documents from a set of large texts using the assumed relations. Then ANGE uses the extracted documents and structures to improve accuracy. ANGE also differs in that it considers two distinct types of documents each with a distinct structure. The work in (Bhole et al. 2007) extracts named entities and relations and presents them as graphs annotated with extracted temporal entities from the same documents. ANGE differs in that it extracts graphs from a class of documents and annotates the extracted graph with entities it extracts from another class of documents. ANGE uses a structural and morphological metric to compute identical entities instead of the Levenshtein distance used in (Finin et al. 2009) to evaluate cross-document coreference in Wikipedia. In (Ravin and Kazi 1999) English names are coreferenced using split and merge heuristics based on the ambiguity level of the names after contextual analysis. ANGE differs in that it uses graph algorithms and considers structural contextual features specific to the subject documents.

narrators	recall	precision	F-score
detected	0.673	0.824	0.741
all	0.632	0.771	0.694

Table 1: Narrator graph accuracy results.

	Narrator detection			Narrator boundary		
	recall	precision	F-score	recall	precision	F-score
No-Cross-doc	0.94	0.81	0.87	0.41	0.95	0.57
Cross-doc	0.65	0.91	0.76	0.93	0.96	0.94

Table 2: Narrator detection in biographies.

	Biography detection			Biography boundary		
	recall	precision	F-score	recall	precision	F-score
all	0.13	0.72	0.21	0.09	0.79	0.16
interesting	0.71	0.84	0.77	0.47	0.87	0.61
in-graph	0.80	0.89	0.84	NA	NA	NA

Table 3: Biography detection results.

6 Results

We evaluated ANGE using three hadith books with several volumes each (Ibn Hanbal 2005; Al Kulayni 1996; Al Tousi 1995), and two biography books (Al-Kassem Al-Khoei ; Al-Waleed Al-Baji) that we obtained from online sources. We report recall and precision as our evaluation metrics. Informally, recall measures whether we detected all correct entities; while precision measures whether we did so without introducing false positives.

The results in Table 1 report the accuracy results for the narrator graph extraction from the hadith books. For each narrator node n, we computed the ratio of the correctly merged nodes against the number of nodes with which n should be merged. The precision metric denotes the average of the ratio of correctly merged nodes against the number of merged nodes for each narrator node.

The "detected" row includes only the narrator nodes detected by ANGE and indicates the accuracy of the graph building routine isolated from the rest of the method and excludes the narrators that ANGE failed to extract. The "all" row includes all narrators and sanads in the text. The recall rate was 63% mainly due to the conservative cycle breaking transformation that splits merged narrator nodes even if they were equivalent. ANGE achieved 82% and 77% precision rates. Upon examination, we discovered that the precision should have been higher due to a conservative decision by the manual annotator to keep several confusing nodes split while ANGE correctly merged them. Further studies should accomodate several manual annotators and report on inter-annotation agreement.

Table 2 shows the accuracy results for narrator detection in biography books computed via manually inspecting a selection of 1,950 narrator names. Without the use of the narrator graph extracted from the hadith documents, ANGE detected 94% of the narrators mentioned in the biography books with 81% precision, and detected 40% of the boundaries of the extracted narrators with 94.5% precision.

With the use of the narrator graph, ANGE coreferenced 65% of the narrator names in the biography books with 91% precision. The rest of the biography narrator names are not found in the hadith books that we used to generate the narrator graph and are therefore not interesting to the narra-

tor graph annotation task. ANGE improved the recall of the boundaries of the detected narrators from 40% to 93% and improved the precision also from 94.5% to 96%. This shows the significant impact of cross-document reconciliation on the accuracy of named entity and relation extraction.

We consider a biography interesting when it contains at least two student/professor names. We computed the results in Table 3 via inspecting a selection of 150 biographies. The recall figures are low when inspecting all biographies. The recall for biography detection improves to 71% from 13% when inspecting the interesting biographies. The recall for biography boundary detection improves to 47% from 9%. Precision also improves significantly. The results should improve when we include more hadith books and build more complete narrator graphs. This is evident when we inspect interesting biographies whose subject narrators exist in the narrator graph extracted from hadith books.

We considered sets of at most ten hadith documents related by content and extracted their narrator graphs using ANGE. We then used the partial narrator graph annotation approach to annotate the resulting graphs. ANGE annotated the nodes of the partial graphs with 80% recall and 89% precision. This is again evidence that cross-document NLP improves entity and relation extraction results significantly. The accuracy of biography boundary detection is not well defined in this task since the partial graph annotation method reports several biographies ranked with a similarity metric.

Acknowledgements. We would like to thank Marwan Zeineddine, Hassen Al-Sadr, and Hussein El-Asadi from the Hadithopedia team for technical discussions, and discussions on the hadith application.

7 Conclusion and future work

We presented ANGE, an annotated narrator graph extraction technique from hadith and biography books that uses morphological features, finite state machines, graph algorithms and cross-document reconciliation and integration. The use of cross-document reconciliation significantly improved accuracy results on named entity and relation extraction tasks. Up to our knowledge, ANGE is the first tool that uses cross-document NLP to improve accuracy results in Arabic NLP tasks. We plan to use cross-document NLP to learn feature vectors that characterize relations such as studentship that was hard to detect in the biography documents without cross-document NLP. We also plan to extend our cross-document approach to other applications.

References

Abuleil, S. 2004. Extracting names from Arabic text for question-answering systems. In *Recherche d'Information et ses Applications (RIAO)*, 638–647.

Al-Jumaily, H.; Martnez, P.; Martnez-Fernndez, J.; and Van der Goot, E. 2011. A real time named entity recognition system for Arabic text mining. *Language Resources and Evaluation* 1–21.

Al-Kassem Al-Khoei, A. *Moajam Rijal Al hadith (Encyclopedia of hadith narrator biographies — in Arabic)*.

Al Kulayni, M. i. Y. 1996. *Kitab al-Kafi*. Taaruf.

Al-Sughaiyer, I. A., and Al-Kharashi, I. A. 2004. Arabic morphological analysis techniques: a comprehensive survey. *American Society for Information Science and Technology* 55(3):189–213.

Al Tousi, M. b. H. 1995. *Al Istibsar*. Taaruf.

Al-Waleed Al-Baji, A. *Al-taadeel wa al-tajreeh liman akhraj 'nh al-Bokhari (The credibility and the scrutinized credibilty for the men of Bokhari — in Arabic)*.

Askari, M. A. 1984. *Maalim al Madrasatayn*, volume 2. Bitha.

Azami, M. M. A. 1991. A note on work in progress on computerization of hadith. *Journal of Islamic studies* 2(1).

Azmi, A., and Bin Badia, N. 2010. iTree - automating the construction of the narration tree of hadiths. In *Natural Language Processing and Knowledge Engineering*.

Benajiba, Y.; Zitouni, I.; Diab, M. T.; and Rosso, P. 2010. Arabic named entity recognition: Using features extracted from noisy data. In *ACL (Short Papers)*, 281–285.

Benajiba, Y.; Diab, M.; and Rosso, P. 2008. Arabic named entity recognition using optimized feature sets. In *Empirical Methods in Natural Language Processing*, 284–293.

Benajiba, Y.; Rosso, P.; and Benedíruiz, J. 2007. ANERsys: An Arabic named entity recognition system based on maximum entropy. 143–153.

Bhole, A.; Fortuna, B.; Grobelnik, M.; and Mladenic, D. 2007. Extracting named entities and relating them over time based on wikipedia. *Informatica (Slovenia)* 31(4).

Buckwalter, T. 2002. Buckwalter Arabic morphological analyzer version 1.0. Technical report, LDC catalog number LDC2002L49.

Finin, T.; Syed, Z.; Mayfield, J.; McNamee, P.; and Piatko, C. 2009. Using Wikitology for Cross-Document Entity Coreference Resolution. In *AAAI Symposium on Learning by Reading and Learning to Read*. AAAI Press.

Ibn Hanbal, A. .-. B. 2005. *Musnad*. Noor Foundation.

Maloney, J., and Niv, M. 1998. TAGARAB: A fast accurate Arabic name recognizer using high-precision morphological analysis. In *Workshop on Computational Approaches to Semitic Languages*.

Ravin, Y., and Kazi, Z. 1999. Is hillary rodham clinton the president? disambiguating names across documents. In *PROCEEDINGS OF THE ACL '99 WORKSHOP ON COREFERENCE AND ITS APPLICATIONS*, 9–16.

Shaalan, K. F., and Raza, H. 2009. NERA: Named entity recognition for Arabic. *JASIST* 60(8).

Shaalan, K. F.; Magdy, M.; and Fahmy, A. 2010. Morphological analysis of ill-formed arabic verbs in intelligent language tutoring framework. In *Applied Natural Language Processing, Florida Artificial Intelligence Research Society Conference*. AAAI Press.

Sipos, R.; Bhole, A.; Fortuna, B.; Grobelnik, M.; and Mladenic, D. 2009. Historyviz - visualizing events and relations extracted from wikipedia. In *European Semantic Web Conference*, volume 5554 of *Lecture Notes in Computer Science*.

Stephanie Strassel, Mark Przybocki, K. P. Z. S., and Maeda, K. 2008. Linguistic resources and evaluation techniques for evaluation of cross-document automatic content extraction. In *International Conference on Language Resources and Evaluation*.

Traboulsi, H. 2009. Arabic named entity extraction: A local grammar-based approach. In *International Multiconference on Computer Science and Information Technology*.

Zaghouani, W.; Pouliquen, B.; Ebrahim, M.; and Steinberger, R. 2010. Adapting a resource-light highly multilingual named entity recognition system to Arabic. In *Language Resources and Evaluation Conference*.

Zhang, Z.; Otterbacher, J.; and Radev, D. R. 2003. Learning cross-document structural relationships using boosting. In *ACM International Conference on Information and Knowledge Management*, 124–130.