Arabase
A Database Combining Different Arabic Resources with Lexical and Semantic Information

Hazem Raafat[1], Mohamed Zahran[2] and Mohsen Rashwan[3]
[1]*Computer Science Department, Kuwait University, Kuwait City, Kuwait*
[2]*Computer Engineering Department, Cairo University, Giza, Egypt*
[3]*RDI, Faculty of Engineering, Cairo University, Giza, Egypt*

Keywords: Natural Language Processing, Arabic Language Resources, Text Similarity Matching, Arabic Language Resources Integration.

Abstract: Language resources are important factor in any NLP application. However, the language resource support for Arabic is poor because the existing Arabic language resources are either scattered, inconsistent or even incomplete. In this paper we discuss the notion of having an integrated Arabic resource leveraging various pre-existing ones. We present a comparison between these resources then we present preliminary fully and semi-automated methods to integrate these resources. This work serves as a bootstrapping for a rich Arabic-Arabic resource with a good potential to interface with WordNet.

1 INTRODUCTION

Language resources have a great impact on the quality of any NLP application. The term "Language resource" refers to any machine readable pieces of information in any format providing information about language being processed. For example, language resources for a certain language can provide the following information for a given word: different senses with definitions and examples for each sense, different word relations like synonyms and antonyms, word morphological analysis and Semantic information.

These information are important for many NLP applications like word sense disambiguation, text similarity, semantic search, text mining, opinion mining, and many others. Fassieh (Attia et al., 2009) is one example on these applications.

A lot of work is done in the field of language resources for many languages like English, French, German and many other European languages, but very few is done to Arabic. Moreover, these few Arabic language resources are limited and not fully developed. Yaseen, et al. (2006) conducted a review on Arabic language resources. A good language resource can be done manually by expert linguists, but such task can take a long time and too much human labor to achieve.

In this paper we examine various Arabic language resources, compare them and apply an algorithm to integrate the scattered information across these different resources into one compact database using a configurable technique between fully and semi-automated methods showing the trade-off them in the integration.

The paper is organized as follows, section 2 discusses related work, while section 3 presents a comparison of different Arabic resources. Section 4 presents the proposed integration methodology. Section 5 presents the database architecture and description. Section 6 presents the evaluation and testing. Section 7 discusses the limitations and future work, and finally we conclude our work in section 8.

2 RELATED WORK

Several attempts have been recorded to enrich the Arabic language resources. Elkateb, et al. (2006) reported on efforts for building a WordNet for Arabic. They followed the methods developed for EuroWordNet (Vossen, 1998). Concepts from WordNet (Princeton University "About WordNet.",

Raafat H., Zahran M. and Rashwan M..
Arabase - A Database Combining Different Arabic Resources with Lexical and Semantic Information.
DOI: 10.5220/0004656702330240
In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval and the International Conference on Knowledge Management and Information Sharing* (SSTM-2013), pages 233-240
ISBN: 978-989-8565-75-4

2010), EuroNet languages and BalkaNet (Tufis, 2004) are used as Synsets in ArabicWordNet. Then some Arabic language specific concepts are translated and added too. Equivalent English entries according to SUMO ontology (Niles and Pease, 2001) are translated and added as well. Finally a bi-directional propagation is performed from English to Arabic and vice versa to generate Synsets. Most of the work is done manually which decreased the coverage and depth of the resulting resource.

Another attempt is bootstrapping an ArabicWordNet using WordNet and parallel corpora (Diab, 2004). She exploits the fact that a polysemous word in one language will have a number of translations in another language. These translations can be clustered based on the word sense proximity using WordNet. Diekema (2004) attempted to build an English-Arabic semantic resource that can be used in CLIR. She used WordNet and various bilingual resources. Attia et al. (2008) built a rich Arabic lexical semantic database based on the theory of semantic field using various Arabic resources.

3 ARABIC LANGUAGE RESOURCES

We started our work by searching for different Arabic language resources, and exploring what information they provide. First, we present the nature of information provided by these resources, and then we present a comparison between these resources.

Below is a list of the information provided by these Arabic language resources.

Morphological information (I_m): The Word morphological analysis, like root, type, gender, and number.

Sense information (I_{se}): Different word senses with gloss, definitions and examples for each sense.

Synset information (I_{sy}) (adapted from WordNet terminology): Different relations between two sets of words. E.g. synonyms and antonyms.

Semantic information(I_{sm}): The Semantic field that the word belongs to together with different semantic relations between semantic field pairs.

Interfacing with WordNet (I_{wn}): Arabic words are linked to their equivalent English words in WordNet.

Each word can have one or more of these information components in what we call the 'information vector' **< I_m, I_{se}, I_{sy}, I_{sm}, I_{wn}>**

Next, we present the resources we found and we give tabular comparison between them in table 2.

The main entry of all resources is the unvocalized word which can have more than one vocalized form. Vocalized forms are classified by their part of speech (POS) into nouns, verbs and particles. Each vocalized form can have its own morphological information and it can have also more than one sense. Each sense can have its sense information, synset information, semantic information and a WordNet interface if applicable.

3.1 King Abdulaziz City for Science and Technology (KACST)

It is an Arabic-Arabic resource in a MYSQL database format (almuajam, 2011). It provides sense and morphological information. It has very limited and incomplete semantic information (Figure1).

3.2 Arramooz

It is an Arabic-Arabic resource available in different formats (SQL, XML and raw text) (Arramooz AlWaseet, n.d.). It provides morphological and sense information (Figure2).

3.3 Arabic WordNet (AWN)

It is an Arabic-English resource in derby database format (Arabic WordNet, 2007). Its main entry is both English and Arabic words. It provides Arabic-Arabic information by mapping and translating the pre-existing English-English information obtained from WordNet, thus interfacing with it. It also provides synset and sense information (Figure3).

3.4 RDI Lexical Semantic Database

It is an Arabic-Arabic resource by RDI available in access format (Attia et al., 2008). It provides semantic information and some sense, morphological information (Figure4).

3.5 RDI Lite Lexicon

This resource is limited. It provides some morphological information and some sense information.

3.6 Alkhalil

It is an Arabic morphological analyser used to add some morphological information when needed in the integration (alkhalil dot net, 2011).

3.7 Arabic Stop Words

It provides Arabic stop words with morphological inflections using a combinations of possible prefixes and suffixes (Arabic Stop Words, 2010). It has 10,389 unique entries.

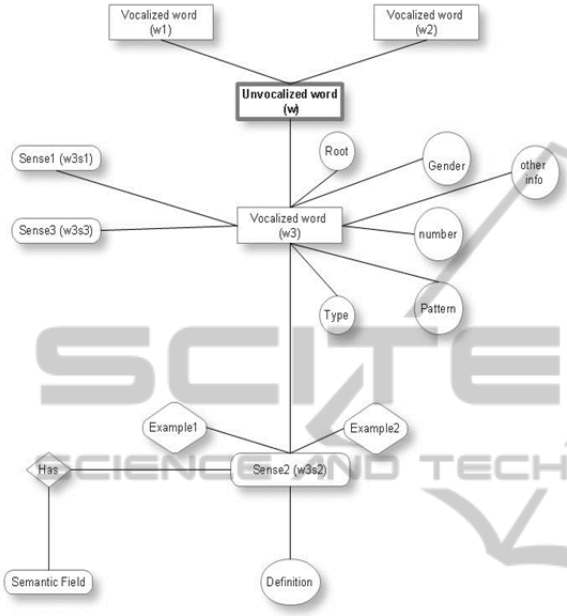

Figure 1: Graphical representation for KACST.

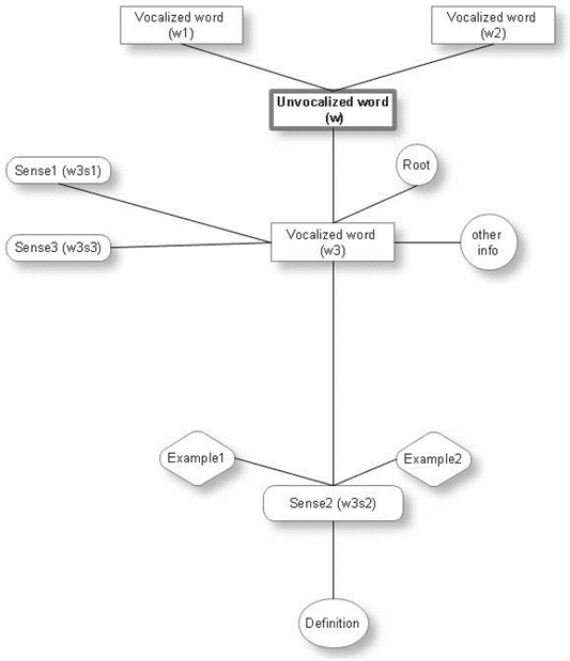

Figure 2: Graphical representation for ARRAMOOZ.

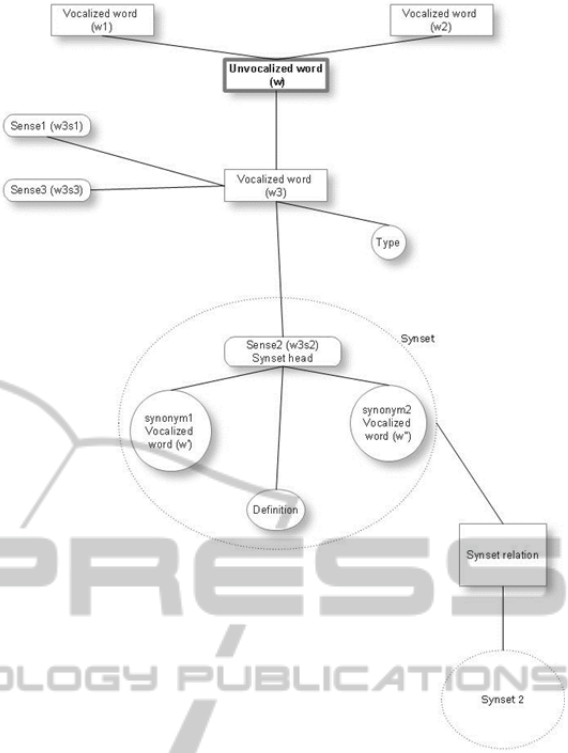

Figure 3: Graphical representation for AWN.

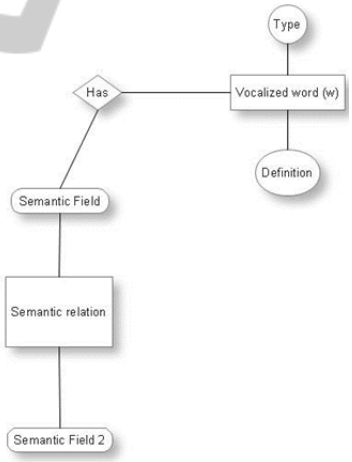

Figure 4: Graphical representation for RDI-LSDB.

4 INTEGRATION METHODOLOGY

The main goal of the integration is to try to allocate an information vector from all available resources for each distinct vocalized word. The integration process is done in four main steps; Analysis, Design, Integration, and Linking.

4.1 Analysis

Carefully analyse each resource to detect its potential Information components. Then we transform all the resources into one format to ease the integration process. We transformed all the resources to MYSQL database format.

4.2 Design

Put a design for the target integrated database given the resources we have. The scalable design of KACST database was the starting point of our integrated database, then we modified it to match the target database design as shown in figure 5. Also table 2 show statistical comparison between the existing and the target database, only the distinct and non-floating entries are counted. Floating entries are words entries with no information components.

4.3 Integration

Design and apply an algorithm to automatically compile these resources together. The following is an overview of the integration algorithm:

For each vocalized word *w* in resource *r*:
- Get (POS) of *w* if provided by *r*, otherwise use Alkhalil to get POS of *w*
- Add *w* to a table corresponding to its POS only if *w* was not added before, otherwise use the existing entry.
- For each information *i* in *r* for *w*
  - Add *i* to *w*
- For each vocalized word *w*
  - Cluster information *i* for *w* by word sense.

'i' refers to an information component.

4.4 Linking

It is the information linking phase in which we cluster the information content for each unvocalized word by its senses.

5 ARABASE ARCHITECTURE AND DESCRIPTION

5.1 Description

The main entry in the integrated resource is the unvocalized word which has more than one vocalized form. These vocalized forms can be nouns, verbs, particles or unclassified. For example consider the unvocalized word (*w*):عين (pronounced *Aien*) it has more than one vocalized form
E.g. (w_1): عَيْن, (w_2):عَيِّن
Each of these vocalized forms can have more than one sense. E.g. w_1 has these two senses. $Sense_1$ عُضْو الإبصار للإنسان وغيره من الحيوان (translated as *Eye*). $Sense_2$ الجاسُوس (translated as *Spy*).

Each sense can have a meaning or a definition, and more than one example to illustrate its usage. Each of these senses can have more than one synonyms.
E.g. $Sense_2$ can have these two synonyms عميل, مراقب. These synonyms form a *Synset*. We choose one of these synonyms and promote it to be the Synset head. We should note that synonyms themselves are in fact vocalized words.

Each Synset can have a semantic field. E.g. $synset_1$ {اسود, داكن, حالك} (Synonyms for the word *dark*) can have the word لون (translated as *color*) as its semantic field. Also the $synset_2$ {ابيض, فاتح, ناصع} (Synonyms for the word *bright*) can have لون as its semantic field.

Synsets can have relations with each other E.g. hyponym, antonym and synonym. E.g. $synset_1$ & $synset_2$ are antonyms.

Semantic fields can have relationships with one another as well.

Entries in the integrated resource are divided among four main tables; Nouns, Verbs, Particles and Not classified. The table 'Not classified' is for words we couldn't specify its (POS) during the integration process because this information is missing from the resource or Alkhalil failed to analyze the word.

Our goal is to compile as much information vector components as possible for each vocalized word *w*. These components will possibly be compiled from different resources ($r_1, r_2 \ldots$) so that the final vector could be:
w :<$r_1(I_m)$, $r_2(I_{se})$, $r_3(I_{sy})$, $r_4(I_{sm})$, $r_5(I_{wn})$>. Where $r_x(I_y)$ is the piece of information *y* allocated from the resource *x*.

5.2 Problems with Integration

Let w_1 and w_2 be two different senses from two different resources for the same word *w*. Since different senses of the same word can have the same vocalized form of the word, therefore w_1 and w_2 have the same vocalized form. So we cannot rely solely on the vocalized word form to distinguish between different words senses, which means that w_1 and w_2 could be the same sense or two different senses. This confusion is a problem in the

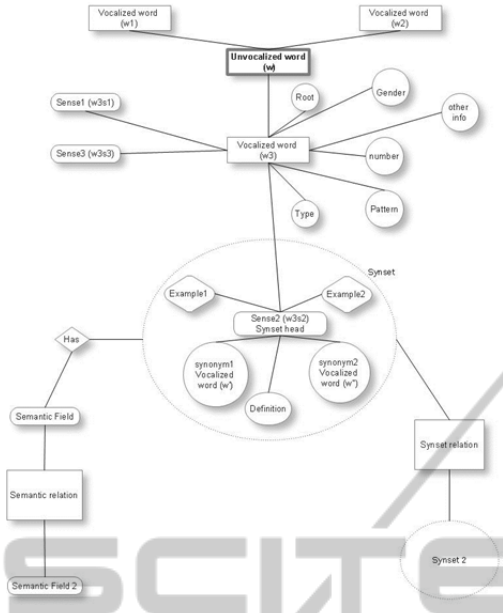

Figure 5: Graphical representation for Arabase.

integration. The following is an example on that confusion:

For the vocalized word *w* عَيْن.

Resource r_1 provides <I_{se}, I_m>

w_1: عَيْن

I_{se}: عُضْو الإبصار للإنسان (Definition for the word *Eye).*

I_m: مفرد, مؤنث, اسم (Some morphological information).

Resource r_2 provides <I_{se}, I_{sm}>

w_2: عَيْن

I_{se}: الجاسُوس (translated as *Spy*).

I_{sm}: الرقيب (Semantic field).

Resource r_3 provides <I_{se}, I_{sm}>

w_3: عَيْن

I_{se}: من جوارح الانسان الخاصة بالبصر (Definition for the word *Eye*).

I_{sm}: عضو من الأعضاء (Semantic field).

Resource r_4 provides <I_{sy}, I_{wn}>

w_4: عَيْن

I_{sy}: عميل, مراقب (Synonyms for the word *Spy*).

I_{wn}: Spy (Corresponding WordNet entry).

We notice that w_1, w_2, w_3 and w_4 have the same vocalized form but w_1 & w_3 have the same sense and w_2 & w_4 have the same sense as well. Our goal is to collect two information vectors one for each sense.

The first is for the sense (*Eye*) < r_1 (I_m), r_1 (I_{se}) r_3 (I_{se}), r_3 (I_{sm})> and the second is for the sense (*Spy*) < r_2 (I_{se}), r_2 (I_{sm}), r_4 (I_{sy}), r_4 (I_{wn}) >.

In order to do so, we have to search in the available information for clues that infer that the information from r_1 & r_3 and r_2 & r_4 belong together. At the integration phase we assume that each information vector *I* comes from resource *r* represents a distinct sense. I.e. r_1 (I), r_2 (I), r_3 (I) and r_4 (I) are totally different and each represents a different sense (i.e. different synsets) for the word *w* and finally at the linking phase we analyse these information and link the related information together.

5.3 Proposed Solutions

We discuss some heuristics we used in the linking phase. **The first**: If two information components belonging to two different resource are linked together then all the information content of these resources are linked together as well.

The second: All words in the same synset, i.e. synonyms, share all the links established by all the synset members. (Except for I_m).

Generally all the sense information (I_{se}) are definitions, which means we can use text similarity algorithms to decide if two given definitions are similar or not, then using the heuristic discussed above we link all the information content of their resources.

If we look closely to the previous example we find that we need to link r_1 & r_3 and r_2 & r_4. We can link r_1 (I_{se}) with r_2 (I_{se}) using text similarity algorithms. Then we decide that r_1 & r_3 are linked together to the same sense.

Generally, text similarity algorithms give a similarity score for a given two texts, but it does not decide if the two texts are similar or not. In this case, to decide if two texts are similar or not using similarity score algorithms we have the following three alternatives.

The first is to use the similarity score when querying the integrated database. We can show the links with their scores or show only the links with confidence exceed a certain value. It turned out that this method can cause problems when using the integrated database in different NLP applications.

The second is to find a threshold such that if the similarity score between two texts is greater than this threshold we label these two texts as similar and not similar otherwise. The main drawback of this method is the false positives (labelling two texts as similar which in fact they are not) are dangerous because the result of the text similarity decision is a clue to link all the information of the two resources these texts come from, which can cause serious confusion.

The third is to do a Configurable semi-automatic linking. There is no doubt that the linking task and the confusion problem are solved using

human labor. But this is very time consuming task. We propose to use text similarity techniques to reduce the time taken by humans to do this task as follows:

For a given two resources r_1, r_2 having the vocalized word w in common. We retrieve all the senses for w per resources r_1 {$sense_1$, $sense_2$ …}, r_2 {$sense_1$, $sense_2$ …}

For each sense s in r_1

    For each sense s` in r_2

        s.calculateSimilarityScore(s`)

    s.sortScoresDescendingly ()

This way each sense has a list of senses sorted by similarity score. When one sense is chosen, all the similar senses will appear sorted by similarity score so there will be no need to scan all the senses and most likely a match will be found in the first *k-best* senses. The more accurate the text similarity algorithm the less k will be before finding a match. We compared two text similarity algorithms in terms of the *k-best measure*; Modified Lesk and latent semantic analysis.

5.3.1 Modified Lesk

Lesk (1986) method is used in word sense disambiguation problems. We modified it to do text similarity based on scoring two texts by counting the number of common words between them. The intuition behind that is that two texts expressing the same meaning usually use similar words. We modified the behaviour of the algorithm by introducing some parameters to be tuned on a development-set. Some of these parameters are Boolean yes/no and others take real values. Boolean parameters are: removing diacritization, removing stop words, stemming the words and using edit distance. Real valued ones: stop word to stop word weight, stop word to non-stop word weight.

5.3.2 Latent Semantic Analysis

LSA (Deerwester et al., 1990) uses a corpus to build a matrix whose rows represent unique words and columns represent each paragraph (paragraphs in our problem are the word's definitions). Singular value decomposition (SVD) is then used to reduce the number of columns while preserving the similarity structure among rows. Texts to be compared are projected on this space and the similarity is calculated based on the cosine of the angle between the two vectors. The intuition behind using a semantic similarity algorithm like LSA is that texts expressing the same meaning are usually semantically similar. The number of dimensions used by LSA is yet to be tuned using a development-set.

6 EVALUATION AND TESTING

We have to do two kinds of evaluations. The first is the depth and breadth coverage evaluation of the integrated resource itself. Breadth is the number of entries found in the database, while depth is the information content for each entry. The second is the evaluation of the linking algorithm.

6.1 The Evaluation of the Integrated Resource (Arabase)

In order to evaluate the database in both depth and breadth coverage. We used a random sample of running Arabic text collected form Wikipedia from different topics, then for each word in the running text we retrieved all the possible word forms from Arabase (referred to as hits) this represents the breadth coverage. For each hit we retrieved all its possible information content (I_m, I_{se}, I_{sy}, I_{sm}, I_{wn}) .This represents the depth coverage. (Table 1).

Table 1: Depth and breadth coverage of Arabase.

	Arabase
#words	2059
#hits	6997
#missed	10
#stopwords	450
Total I_m	2546
Total I_{se}	6055
Total I_{sm}	4064
Total I_{sy}	871
Total I_{wn}	871
Avg. I_m per hit	0.3639
Avg. I_{se} per hit	0.8654
Avg. I_{sm} per hit	0.8653
Avg. I_{sy} per hit	0.1245
Avg. I_{wn} per hit	0.1245

6.2 The Evaluation of the Linking Algorithm

We examined some words and manually link information components together based on I_{se} resulting in 184 links. We used 134 of them as a development-set to tune the parameters for both Lesk and LSA and we used the remaining 50 for testing.

Table 2: Shows a statistical comparison between the existing and the final integrated resources.

	KACST	Arramooz	AWN	RDI-LSDB	RDI-Lite	Integrated Resource (Arabase)
#Nouns	42,732	27,272	11,564	15,095	N/A	81,977
#Verbs	18,054	9,866	3,226	8,254	N/A	35,400
#Particles and stop-words	171	N/A	N/A	N/A	N/A	10,389
#Not classified	N/A	N/A	1,290	27,883	7,354	30,651
#Total entries	60,957	37,138	16,080	51,232	7,354	158,417
#Meaning definitions	80,512	19,053	N/A	33,394	13,938	146,897
#Examples	5,232	N/A	N/A	N/A	2,554	13,388
#Semantic fields	247	N/A	N/A	5,039	N/A	5187
#Semantic field relations	N/A	N/A	N/A	292,910	N/A	292,799
#Semantic relations types	N/A	N/A	N/A	19	N/A	19
#Synsets relations types	N/A	N/A	22	N/A	N/A	22
#Synset relations	N/A	N/A	145,655	N/A	N/A	145,655
Information components	I_m, I_{se}, I_{sm}	I_m, I_{se}	I_{se}, I_{sy}, I_{wn}	I_m, I_{se}, I_{sm}	I_m, I_{se}	$I_m, I_{se}, I_{sy}, I_{sm}, I_{wn}$

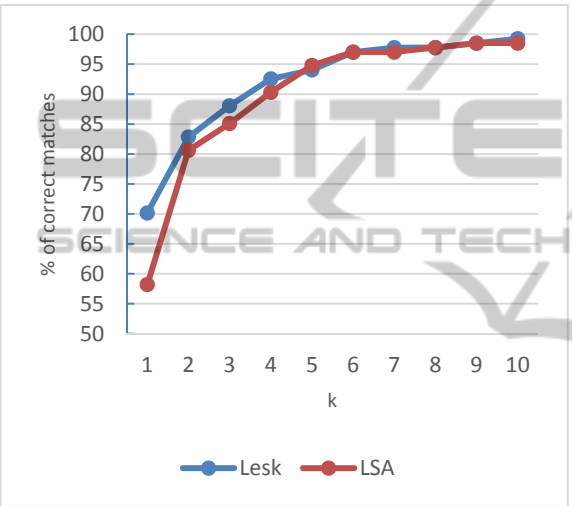

Figure 6: The performance of Lesk VS. LSA on the development-set using the k-best measure.

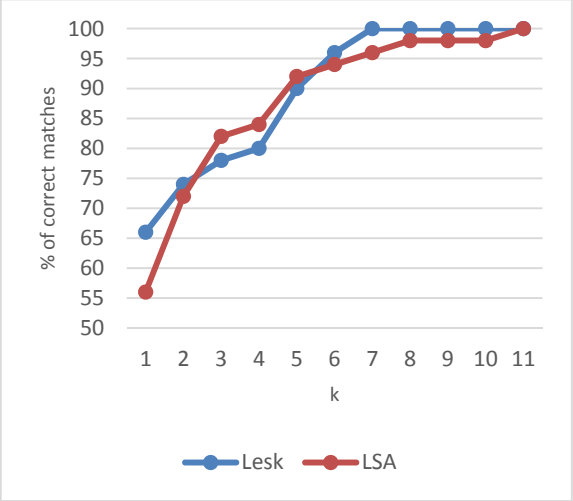

Figure 7: The performance of Lesk VS. LSA on the test-set using the k-best measure.

Figure 6 shows the performance of Lesk VS. LSA on the development-set using the k-best measure. The vertical axis is percentage of correct matches sorted with a rank less than or equal to k. At k=10 both Lesk and LSA didn't yet saturate and reach 100%. LSA saturates after k=11, while Lesk saturates after k=17.

Figure 7 shows the performance of Lesk VS. LSA on the test-set using the k-best measure. After tuning both Lesk and LSA these parameters that gave best performance on the development-set:
Lesk: Removing diacritization, edit distance: True.
Removing stop words, stemming: False
Stop word to stop word weight=0.01
Stop word to non-stop word weight=0.01
LSA: We used genism API (Rehurek, R., 2010) on a 445MB training corpus collected from (ksucorpus, 2013). The tuned number of dimensions=100.

7 LIMITATIONS AND FUTURE WORK

Since the integration work here is automated with adjustable human interaction, the algorithm is liable to some errors that can be solved manually.

Below are some of the limitations involved in our approach:

Classifying the words by POS is liable to errors when using Alkhalil. If it failed to get the POS, the word is given the label 'Not classified.

N-gram entries (entries with more than one token) are classified according to the POS information given in the resource. If no such information is found we classify it according to the first token. If failed we label it as 'Not classified'.

Poorly diacritized words can confuse the morphological analyser, which ends up with more than one morphological analysis for the same word.

In these cases the first analysis is taken, which can be erroneous approach but can be fixed manually. We can also choose not to provide morphological information for such words.

The linking algorithm is limited to linking resources based on the sense information (I_{se}). If any resource does not have this information component and its synset has no other words, then it will not be linked by our linking algorithm.

We can enrich Arabase by linking it with WordNet. Such that each Arabic sense is linked with its corresponding English one in WordNet. Currently the only interface is the integrated entries from ArabicWordNet.

8 CONCLUSIONS

We compared different Arabic resources examining their points of strength and weakness. Then we presented a framework that can be used to compile pieces of Arabic language information scattered across these resources into a single resource. We showed the trade-off between fully automated and manual methods. Full automation will decrease significantly the human effort, thus saving time and man-power at the expense of decreasing the accuracy and consistency of the resulting resources. We showed the compromise between both methods can result in an acceptable accuracy and consistency with minimal human efforts.

REFERENCES

alkhalil dot net, 2011. KACST Available at: http://sourceforge.net/projects/alkhalildotnet/. [Accessed 23 June 2013].

almuajam, 2011. Arabic Interactive Dictionary Project. Available at: http://sourceforge.net/projects/almuajam/ [Accessed 23 June 2013].

Arabic Stop Words, 2010. Available at: http://arabicstopwords.sourceforge.net/. [Accessed 23 June 2013].

Arabic WordNet, 2007. *A multi-lingual concept dictionary,* Available at: http://awnbrowser.sourceforge.net/. [Accessed 23 June 2013].

Arramooz AlWaseet*: Arabic dictionary for morphology*. Available at: http://arramooz.sourceforge.net/. [Accessed 23 June 2013].

Attia, M., Rashwan, M., Al-Badrashiny M., 2009. 'Fassieh; a Semi-Automatic Visual Interactive Tool for the Morphological, PoS-Tags, Phonetic, and Semantic Annotation of the Arabic Text', *IEEE Transactions on Audio, Speech, and Language Processing (TASLP): Special Issue on Processing Morphologically Rich Languages.*

Attia, M., Rashwan, M., Ragheb, A., Al-Badrashiny, M., Al-Basoumy, H. & Abdou, A., 2008. 'A Compact Arabic Lexical Semantics Language Resource Based on the Theory of Semantic Fields', *Advances in Natural Language Processing,* LNCS vol. 5221, pp 65-76.

Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., & Harshman, R., 1990 'Indexing by Latent Semantic Analysis'. *Journal of the American society for Information science,* vol. 41, no. 6, pp. 391-407.

Diab, M., 2004. 'The Feasibility of Bootstrapping an Arabic WordNet leveraging Parallel Corpora and an English WordNet', *Proceedings of the Arabic Language Technologies and Resources, NEMLAR, Cairo.*

Diekema, A.R., 2004. 'Preliminary Lexical Framework for English-Arabic Semantic Resource Construction',. *Semitic '04 Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages* , Stroudsburg, PA, .pp. 10-14.

Elkateb, S., Black, W., Rodríguez, H., Alkhalifa, M., Vossen, P., Pease, A., & Fellbaum, C., 2006. 'Building a WordNet for Arabic'*, Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006)*, Genoa, pp. 29-34.

ksucorpus, 2013. King Saud University Corpus of Classical Arabic. Available at: http:// ksucorpus.ksu.edu.sa/ar/. [Accessed 23 June 2013].

Lesk, M., 1986. 'Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone', *Proceedings of the 5th annual international conference on Systems documentation SIGDOC '86,* Toronto, pp. 24-26.

Niles, I. & Pease, A., 2001. 'Towards a standard upper ontology'. *Proceedings of the International Conference on Formal Ontology in Information Systems FOIS '01*, Ogunquit, Maine, pp. 2-9.

Princeton University "About WordNet." 2010. *WordNet*. Princeton University. Available at: http://wordnet.princeton.edu. [Accessed 23 June 2013].

Rehurek, R. & Sojka, P., 2010. 'Software Framework for Topic Modelling with Large Corpora', *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, Valetta, pp. 46-50.

Tufis, D. (ed.), 2004. Special Issue on the BalkaNet project. *Romanian Journal of Information Science and Technology*, vol.7, no. 1-2

Vossen, P., 1998. 'Introduction to EuroWordNet', *Computers and the Humanities,* vol. 32, no. 2-3, pp. 73-89.

Yaseen, M., Attia, M., Maegaard, B.... Rashwan, M., et. al., 2006. 'Building Annotated Written and Spoken Arabic LR's in NEMLAR Project' *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006)*, Genoa, pp. 533-538.