Performing Fusion of News Reports through the Construction of Conceptual Graphs

Joel Azzopardi
Department of Computer Science and AI
Faculty of Science
University of Malta
jazz018@um.edu.mt

ABSTRACT

As events occur around the world, different reports about them will be posted on various web portals. Different news agencies write their own report based on the information obtained by its reports on site or through its contacts – thus each report may have its own 'unique' information. A person interested in a particular event may read various reports about that event from different sources to get all the available information. In our research, we are attempting to fuse all the different pieces of information found in the different reports about the same event into one report – thus providing the user with one document where he/she can find all the information related to the event in question. We attempt to do this by constructing conceptual graph representations of the different news reports, and then merging those graphs together. To evaluate our system, we are building an operational system which will display on a web portal fused reports on events which are currently in the news. Web users can then grade the system on its effectiveness.

1. INTRODUCTION

If one browses the news sites on the World Wide Web (WWW) such as Reuters [1], or AFP [2], he can find news reports of events which have happened only minutes before the time of reading. Each event that occurs can be found reported by different authors on different news sites. The different reports on the same event usually contain the same back-bone of the main sub-events but different fine details. Therefore, for a user to get the whole picture of that event with all the different details incorporated with it, he/she would have to read various reports on the same event. Unfortunately, most of the information in each report would be present in the other reports as well. Thus to get all the fine details, the user has to endure reading repeated information.

[1]http://today.reuters.com
[2]http://www.afp.com/english/news/?pid=stories

We attempt to address the above issues in our research by constructing logical representations of the different news reports, and then merging those structures which are representing reports on the same event. The merged structure would then contain all the information obtained from the different reports. This merged structure would then be presented back to the user as a 'fused' report for that event with unnecessary repetition of the same information.

The news reports are represented using structures similar to a conceptual graphs, and the construction of these conceptual graphs is performed using surface based approaches only – i.e. we will not be using approaches which require deep semantic analysis of the text or a knowledge base. In our opinion, the use of surface-based approaches is more appropriate to news reports since they are less computationally expensive to implement and execute, and also new names and terms crop up in the news every day – otherwise the knowledge base will have to be updated regularly. Moreover surface-based approaches will enhance the portability of our system across different domains since unlike knowledge bases, they can be more general and not limited to particular domains.

Within our research, we are also attempting to evaluate our approach in terms of helpfulness to the user.

The structure of the remaining part of this report will be as follows: in the next section, we describe related work done or currently being done in the same area as our research. Then in the following section, we will describe the methodology implemented in our system. A description of our intended method of evaluation will follow in the proceeding section, and in the final section we give our conclusions and our plans for future work.

2. LITERATURE REVIEW

The aim of Document Fusion is to produce a fused report which will contain all the relevant information from the different source documents without repetition ([7], [1], [8]). The reasons behind the need for Document Fusion are various, namely:

- Any document/report from any particular source is never fully objective. On the other hand, a document built from multiple sources produces more objectivity ([14]),

- Reports from different sources on the same event may contain different information – in fact reports from different sources may agree with each other, contradict each other or add new information ([8], [7], [1]).
- Information Fusion may also help in tracking the developments to reports over time ([8], [14]).

The steps involved in document fusion are ([8], [1], [7], [13]):

1. The segmentation of each document into different segments,
2. The logical representation of each segment so that each segment may be compared to segments from other documents,
3. The construction of the 'fused' document.

Information Fusion is most commonly applied to textual news reports ([8], [1], [7]). However, we also encountered examples of the application Information Fusion on video news reports ([14]), and also on search engine result listings ([13]).

Information Fusion involves the segmentation of the different documents and then building relationships between segments from different documents. In its definition of the Cross-Document Structure [8] describes relationships which may occur between *paragraphs*, *sentences*, *phrases* and even *individual words*.

These different levels of granularity present different issues. [7] goes to the coarse side of the spectrum and segments the documents into paragraphs. According to [7], the use of paragraph segments simplifies matters since paragraphs are context-independent and hence fused reports built from paragraphs taken from different sources will be more readable. Furthermore, [7] argues that within the context of news reports, paragraph units are not too coarse since the paragraphs within news reports do not usually contain more than 3 sentences.

In direct contrast to the use of paragraph segments in [7], [1] claims that even sentence segments are too coarse and each sentence may contain more than one theme. Thus the construction of the fused report from sentence units may lead to repetition. [1] recommends the break-down of sentences into phrases – this eliminates the repetition, but then brings forward the need to do sentence re-generation to make the fused report readable.

In our opinion, performing fusion with paragraph segments will need relatively less processing than fusion with finer granularities. However, paragraphs are too coarse for fusion, and inevitably if fusion is to be made from paragraph segments repetition will inevitably occur.

On the other hand, breaking down the sentences into phrases will necessitate the need for sentence generation after fusion has been done to ensure readability of the output document.

We attempt to do fusion using sentence-size segments. To avoid the problem of repetition in the case of sentences containing more than one themes, we will give a priority to the shorter sentences to be used in the final 'fused' document since these are the most likely to contain only one theme.

To build the relationships between the segments, the segments are represented using a graph representation ([8], [1]) unless the segments are coarse as in the case of [7]. A particular representation which is of interest to us is the *DSYNT* structure which is described in [1]. The *DSYNT* is a graph structure whereby a node is built for each phrase, and a phrase consists of a verb with 2 nouns.

A representation of concepts and relationships between concepts which is quite similar to the *DSYNT* structures described previously but is more evolved is the *Conceptual Graphs* Representation. Conceptual Graph representation is a representation which is precise but also human readable ([10]). It consists of concept nodes which represent entities, states, attributes and events, and relation nodes which describe the relationships between the different concepts ([10], [4]).

In our research, the ideal representation would be *Conceptual Graphs*. However, the construction of conceptual graphs from raw text would require semantic information and the support of a knowledge base as described in [11]. Since we try to adhere to surface-based approaches, our aim is to build structures resembling conceptual graphs as much possible but using surface-based approaches only.

In the construction of graph representation for text, the concepts are first extracted from the text to form the graph nodes, and then the relationships between the concepts are built ([6], [3], [9], [3]). Concepts can be defined by certain pre-defined phrases or by the extraction of proper names ([6], [3]), or otherwise they can be extracted by the use of heuristic rules ([9], [3]).

In our research, we have used a combination of both approaches listed above used for concept extraction. A Part-Of-Speech tagger is used to tag the text, and then heuristic rules are used to extract the concepts from the text. Over and above that, simple extraction of proper names is also done, and any proper names which have not been inserted as concepts in the previous step, will be inserted now.

We have encountered various approaches to the construction of relationships between different concepts. On one hand, there are approaches which build relationships based on the semantic meaning of the concepts and the words in the text ([6], [11], [12]). A case of interest is that described in [11]), whose system uses a lexicon of word canonical graphs to build all the possible graphs for each sentence. Then, a semantic knowledge base is used to reject those sentence graphs which do not make sense semantically. [12] does not build only relationships according to semantic meanings, but also builds relationships between those words or phrases which have high frequencies of co-occurrence within text windows of pre-defined size.

On the other hand, we have more surface-level approaches which make use of heuristic rules applied on the Part-Of-Speech tags (produced by a parser) corresponding to the different words. Examples include [9] which build *Noun-*

Verb-Noun tuples, and [1] which builds the *DSYNT* structure described previously.

Another surface-level approach which is quite different to those described above is that described in [3]. In this case, there are two type of constructed relationships between concepts. These are:

- **Named Relations** – each such relationship consists of 2 concepts and a relationship name. Such relationships are extracted using heuristic patterns – e.g. the text '*president of*' if used to extract a named relationship between '*George Bush*' and '*United States*' from the following sentence: '*George Bush, president of the United States*'.
- **Unnamed Relations** – each such relationship consists of two concepts and a relationship strength. These relationships are built by finding sets of concepts which have a high co-occurrence rate within the same sentences.

In our research, we are trying to use surface-based approaches as much as possible. Therefore, the use of knowledge bases to construct relationships between concepts would go against our approach and also would prove costly, and probably also limit the domain in which our system can operate. We think that the construction of *DSYNT* structure or *NVN* tuples is quite straightforward since the text would have already been tagged previously for the concept extraction. On the other hand, the approaches described in [3] are very interesting and can be applied as well.

3. METHODOLOGY

Our system performs the construction of fused news reports in the following steps:

1. The downloading of the news reports from different sources via RSS feeds.
2. The clustering the documents according to their content,
3. The construction of conceptual representations for each document,
4. The merging of conceptual representations for documents within the same cluster,
5. The construction of the fused document.

As we mentioned in the previous section, one of the main issues in information fusion is to avoid repetition. In our case, this is handled during the merging of the conceptual structures used for representations.

3.1 Downloading of News Reports

Within this part, the RSS feeds from a number of pre-defined sources are downloaded. For each downloaded RSS record, the system first checks if the corresponding news report has been downloaded already. If it has not yet been downloaded, the news report will be downloaded, filtered from the surrounding HTML code, and stored within an XML file together with the other details specified in the RSS record.

3.2 Document Clustering

The main 'problem' within the task of document clustering is that the number of final document clusters is not known beforehand.

To tackle this problem, we adapted a technique similar to that described in [5]. We first index the documents, remove the stop-words from the indexes and assign term weights to the index terms using *tf-idf* measure. Then the similarity of that documents with each cluster is calculated and if there exists a cluster with a similarity higher than a pre-defined threshold, that document is placed within that cluster.

3.3 Building Conceptual Representations for each Document

The logical representation built within this section is an entity-relation structure whereby we define the entities (the 'objects' within the document) and the relations / actions occurring between these entities. Noun entities form the entities, and verb entities describe the relations between these entities.

The steps followed by the system to construct the conceptual graph are as follows:

1. The contents of the document in question are read and are tagged using a POS Tagger.
2. The noun entities are extracted from the document's contents.
3. The complex noun entities within the document i.e. entities formed from 2 or more 'simple' noun entities are extracted.
4. The verb entities are extracted. These will be the names of the named relations between 1 or 2 entities.
5. The relationships between the noun entities (concepts) are built.
6. The relations are grouped into series of relations which lead from one to other.
7. Co-referring noun entities are grouped together.

The following sub-sections contain a more detailed description of each step.

3.3.1 Tagging the Document's Contents

The document's contents are tagged using Brill's Part of Speech tagger [2].

After Brill's Part of Speech tagger has been employed to tag the text, some corrections are applied to the resulting tagged text. Our system reads a text file which contains a list of tokens each with its own corresponding tag and assigns all occurrences of these tokens within the document's text to that tag.

Finally, the named entities are identified and the tags of the tokens which make up these names are set to be of type 'PROPER NOUN'. Named entity extraction is performed

by identifying those tokens which start with an upper-case letter but do not occur at the start of a sentence. Once those tokens have been identified, the tokens which occur at the start of sentences are matched with the list of names extracted previously, to check whether they also form part of names.

3.3.2 Extracting the Noun Entities

Noun entity extraction is performed by applying the following rules to the tagged text:

```
<Noun-Entity> = [<Determiner>] (<Adjective>)* (<Noun>)
```

In other words, a Noun Entity must consists of at least one noun token, any number of adjective tokens preceding that noun token, and possibly a Determiner at the start of the noun entity object.

In cases where a sequence of adjective tokens is found without a noun token at the end, the last adjective in the sequence is considered to be a noun token. Occurrences of such sequences are due to erroneous tagging of text by the POS tagger.

3.3.3 Building the Complex Noun Entities

Complex noun entities are composed of two or more 'simple' noun entities which are linked together by a preposition, or a conjunction or disjunction. For example if we have the phrase 'the president of Iraq'; within the step described in section 3.2, we identified 'the president' and 'Iraq' as separate noun entities. However, it is quite obvious that 'the president of Iraq' is referring to a single entity. By generalization of this example, we can therefore build complex noun entities using the rule:

```
<Complex-Noun-Entity> = <Simple Noun Entity>
                        <preposition>
                        <Simple Noun Entity>
```

Another feature of complex noun entities, as used by our system, is their composite nature – i.e. a complex noun entity may be built using other complex noun entity objects. In fact, the actual heuristic rule which we used in building complex noun entity objects is as follows:

```
<Complex-Noun-Entity> = <Simple Noun Entity>
                        <preposition>
      (<Simple Noun Entity> | <Complex Noun Entity>)
```

Note from the above rule that we assume complex noun entities to be right-associative. This is illustrated in the following example.

Imagine that we have the phrase: '*The leader of the terrorist organization in Iraq*'. The simple noun entities within this phrase are: '*The leader*', '*the terrorist organization*' and '*Iraq*'. According to the heuristic rule we used to build the complex noun entities, we first build the complex noun entity shown in Figure 1.

Then we build the final complex noun entity shown in Figure 2.

3.3.4 Extracting the Verb Entities

Verb Entities are used as names for the relationships between noun entities. These are extracting by applying the

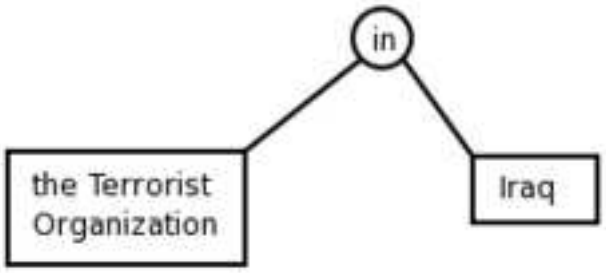

Figure 1: Representation of '*the Terrorist Organization in Iraq*'

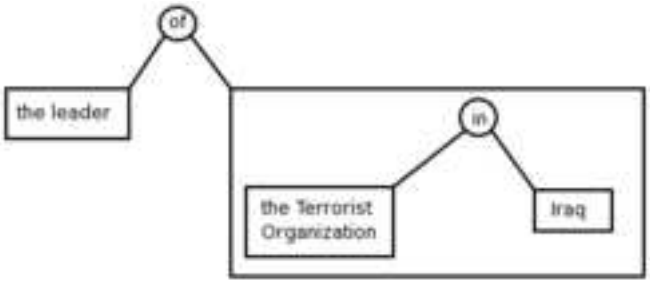

Figure 2: Representation of '*the leader of the Terrorist Organization in Iraq*'

following rule to the tagged text:

```
<Verb-Entity> = (<Adverb>)* [<Auxiliary>] (<Adverb>)*
(<Verb>)*
```

3.3.5 Building the Relationships between the Noun Entities (Concepts)

A Named Relation consists of a verb entity, which provides the name for the relation, and one or two noun entities, between which the relation is defined.

Named Relations may be subdivided into two groups, namely:

- **binary relation** – where the verb entity involved is transitive and the relation is defined between two noun entities. For example, in the phrase '*John kicked Mary*', we have the relation '*kicked*' between '*John*' and '*Mary*',
- **unary relation** – where there is only one noun entity object involved since the verb 'defining' the relation is intransitive. For example, in the phrase '*John died*', we have the 'relation' '*died*' and only the noun entity '*John*' is involved.

Within this sub-section, we are involved in the extraction of these types of entities. To perform this extraction, the system traverses the list of sentence phrases (whose creation is described is section 3.5), and builds the named relations by applying the following heuristic rules in the order given below:

1. `<Named Relation> = <Noun Entity>`

```
                        <Verb Entity>
                        <Noun Entity>

2.  <Named Relation> = <Noun Entity>
                        <Verb Entity>
                        <Phrase Delimiter>

3.  <Named Relation> = <Verb Entity>
                        <Noun Entity>
                        <Phrase Delimiter>
```

If during the construction of the Named Relations, the system finds a temporal entity (a date, or a time string) immediately before or after the entities forming the relations, it attaches this temporal entity to the relation being constructed as an indication of the time/date during which that relation was established.

3.3.6 Identifying Relations Leading from one to another

Within the same sentence phrase, the system may find more than one different named relations. If this is the case, and there is a noun entity which forms part of two relations, those relations are set to 'related' to each other – in the sense that one relation leads off from the other.

For example, consider the sentence, '*The attack hindered the work being done in the country.*' From this sentence, we may extract two named relations – namely '*The attack hindered the work*', and '*the work being done in the country*'. Since there is the noun entity '*the work*' being used in both relations, these two relations are set to be 'related' to one another.

3.3.7 Clustering those noun entities which are referring to the same object

Within a document, different noun entity objects refer to the same real-world object. For example, the noun entities '*the president of the United States*', '*George Bush*', '*the former governor of Texas*' are all referring to the same real-world object – namely George Bush, who is the president of the United States at the time of writing.

Our system attempts to cluster together those noun entities which are 'co-referent' so that we will have as much as possible a one-to-one relation of entities within the conceptual structure constructed to real-life objects. In this way, certain operations, such as the retrieval of all relations which concern a particular object, are greatly facilitated.

This clustering is done in two parts. In the first part, a set of 'significant' tokens is constructed for each noun entity object where 'significant' tokens within a noun entity are those tokens which are the actual nouns. Those noun entities which have equivalent sets of 'significant' tokens are clustered together as co-referent.

The second part of the noun entity clustering utilizes the list on un-named relations whose extraction was described in section 3.5. In this part, the system traverses the list of un-named relations, and for each pair of noun entities, it identifies the two noun entity clusters which contain each noun entity in question, and merges these two clusters together.

3.4 Merging Conceptual Graphs

In the previous section, we described the procedure we used to construct the conceptual graph representation for each document. Now, we need to identify those entities and relations which are common across the different documents of the same cluster.

This *merging* task may be sub-divided into two sub-tasks:

1. The clustering of those noun entity and verb entity objects which are 'synonyms',
2. The clustering of co-referring relations.

3.4.1 Clustering Noun-Entity and Verb-Entity Objects

A similar process is used to cluster co-referring noun-entity and verb-entity objects.

The tokens of each noun-entity and/or verb-entity object are weighted using the *tf-idf* measure based on the inverted index for the document cluster. Those tokens which have a *normalized* weight of 0.5 or greater are considered to be the '*defining*' tokens for that entity object.

When comparing two entity objects, an *intersect* list of defining tokens is extracted, as well as the *difference* list of defining tokens for those two entity objects. The members of the *intersect* list contribute to the equivalence score of those two tokens, whilst the members of the *difference* list hinder this equivalence score. The two entity objects are considered to be co-referring if their equivalence score exceeds a certain threshold.

3.4.2 Clustering the Relations

The approach to the relations' clustering is based on the premise that two relations are clustered if they share at least a *co-referring* verb entity and a *co-referring* noun entity between them, or two *co-referring* noun entities between them.

Two relations are considered to share a *co-referring* verb entity or a *co-referring* noun entity if one of the relations contains a verb (or a noun) entity which forms part of the same verb-entity (or noun-entity) cluster as a verb (or a noun) entity from the other relation.

3.5 Building the Fused Report

Once the previous step has been completed, we end up with a list of conceptual structures representing the concepts and the relations between them. Since these relations have been 'merged' together, we now have a list of 'unique' relations between concepts which represent the different information found in the different reports. The final step involved in the construction of the 'fused' report is to have the system select those sentences which contain the relations represented in conceptual form, and ensure that no relation will be represented in more than one sentence within the final 'fused' document.

4. EVALUATION

Since Document Fusion is a relatively unexplored field, we have not yet encountered any data corpus which provides sample fused documents for clusters of input documents. The information fusion systems described in [13] and [7] use human assessors to evaluate their results. On the other hand, [1] evaluates only similarities found across different documents – this is done by comparing the similar relationships extracted by their system with those extracted by human judges.

To evaluate our approach to Document fusion, we decided to build a News Document Fusion system which will be available on the WWW. This Document Fusion system uses RSS feeds from different sources to download news reports as they are published. The downloaded reports would then be clustered together according to the event they are reporting. Document Fusion would then be applied on the documents within each cluster and the 'fused' document produced will be presented to the user on the WWW site.

To evaluate the Document Fusion system, each fused report on the WWW site will contain links to the original reports, as well as a form where each user can rate that fused report based on the inclusion of all the unique information found in the source reports and the inclusion only once of all the repeating information.

5. CONCLUSION AND FUTURE WORK

Although we do not yet have evaluation results for our system, we are optimistic that our approach is a promising one. The fact that our system is implemented using surface-level approaches only greatly expands the domain across which our system can be operated.

Moreover the representations we use in our approach simplifies the adaptation of our system to other purposes like Topic Tracking and Information Filtering. In fact, in the foreseeable future, we intend to adapt our approach to Topic Tracking and Information Filtering. We intend also to incorporate User Modelling so that a user will be shown a 'fused' report which suits that particular user – i.e. this 'fused' report will contain very few (if any) details on the information that the user already knows, and concentrates on the 'new' details which the user is still to learn about.

6. REFERENCES

[1] R. Barzilay, K. R. McKeown, and M. Elhadad. Information fusion in the context of multi-document summarization. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 550–557, Morristown, NJ, USA, 1999. Association for Computational Linguistics.

[2] E. Brill. A simple rule-based part of speech tagger. In *Proceedings of the third conference on Applied natural language processing*, pages 152–155, Morristown, NJ, USA, 1992. Association for Computational Linguistics.

[3] R. Byrd and Y. Ravin. Identifying and extracting relations in text. In *NLDB 99 – 4th International Conference on Applications of Natural Language to Information Systems*, Klagenfurt, Austria, 1999.

[4] M. Chein and M.-L. Mugnier. Conceptual graphs: fundamental notions. In *Revue d'Intelligence Artificielle, Vol. 6, no. 4, 1992*, pages 365–406, 1992.

[5] B. Larsen and C. Aone. Fast and effective text mining using linear-time document clustering. In *KDD '99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 16–22, New York, NY, USA, 1999. ACM Press.

[6] I. Mani and E. Bloedorn. Multi-document summarization by graph search and matching. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, Menlo Park, California, USA, 1997. AAAI Press.

[7] C. Monz. Document fusion for comprehensive event description. In *Proceedings of the workshop on Human Language Technology and Knowledge Management*, pages 1–8, Morristown, NJ, USA, 2001. Association for Computational Linguistics.

[8] D. R. Radev. A common theory of information fusion from multiple text sources step one: cross-document structure. In *Proceedings of the 1st SIGdial workshop on Discourse and dialogue*, pages 74–83, Morristown, NJ, USA, 2000. Association for Computational Linguistics.

[9] K. Rajaraman and A.-H. Tan. Knowledge discovery from texts: a concept frame graph approach. In *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*, pages 669–671, New York, NY, USA, 2002. ACM Press.

[10] J. F. Sowa. Semantics of conceptual graphs. In *Proceedings of the 17th annual meeting on Association for Computational Linguistics*, pages 39–44, Morristown, NJ, USA, 1979. Association for Computational Linguistics.

[11] J. F. Sowa and E. C. Way. Implementing a semantic interpreter using conceptual graphs. *IBM J. Res. Dev.*, 30(1):57–69, 1986.

[12] N. Stokes and J. Carthy. First story detection using a composite document representation. In *HLT '01: Proceedings of the first international conference on Human language technology research*, pages 1–8, Morristown, NJ, USA, 2001. Association for Computational Linguistics.

[13] T. Tsikrika and M. Lalmas. Merging techniques for performing data fusion on the web. In *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*, pages 127–134, New York, NY, USA, 2001. ACM Press.

[14] Y. Zhai and M. Shah. Tracking news stories across different sources. In *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*, pages 2–10, New York, NY, USA, 2005. ACM Press.