Unsupervised Content Discovery from Concise Summaries

Horacio Saggion
Universitat Pompeu Fabra
Department of Information and Communication Technologies
TALN Group
C/Tanger 122 - Campus de la Comunicación
Barcelona - 08018
Spain

Abstract

Domain adaptation is a time consuming and costly procedure calling for the development of algorithms and tools to facilitate its automation. This paper presents an unsupervised algorithm able to learn the main concepts in event summaries. The method takes as input a set of domain summaries annotated with shallow linguistic information and produces a domain template. We demonstrate the viability of the method by applying it to three different domains and two languages. We have evaluated the generated templates against human templates obtaining encouraging results.

1 Introduction

Our research is concerned with the development of techniques for knowledge induction in the field of text summarization. Our goal is to automatically induce the necessary knowledge for the generation of concise event summaries such as the one shown in Figure 1. This kind of summaries, which can be found on the Web and in text collections, contain key information of the events they describe. Previous work in the area of text summarization (DeJong, 1982; Oakes and Paice, 2001; Saggion and Lapalme, 2002) addressed the problem of generating this type of concise summaries from texts, relying on information extraction and text generation techniques. These approaches were difficult to port to new domains and languages because of the efforts needed for modelling the underlying event template structure. In this paper we propose a method for learning the main concepts in domain summaries in an unsupervised iterative procedure. The proposed algorithm takes a set of unannotated summaries in a given domain and produces auto-annotated summaries which can be used for training information extraction and text generation systems. Domain adaptation is essential for text summarization and information extraction, and the last two decades have seen a plethora of methods for supervised, semi-supervised, and unsupervised learning from texts.

2001 August 24: **Air Transat Flight 236** runs out of **fuel** over **the Atlantic Ocean** and makes **an emergency landing** in **the Azores**. Upon landing some of **the tires** blow out, causing **a fire** that is extinguished by **emergency personnel** on **the ground**. None of **the 304 people** on **board the Airbus A330-200** were seriously injured.

Figure 1: Summary in the aviation domain annotated with chunks

For example, in (Li et al., 2010) clustering is applied to generate templates for specific entity types (actors, companies, etc.) and patterns are automatically produced that describe the information in the templates. In (Chambers and Jurafsky, 2009) narrative schemas are induced from corpora using coreference relations between participants in texts. Transformation-based learning is used in (Saggion, 2011) to induce templates and rules for non-extractive summary generation. Paraphrase templates containing concepts and typical strings were induced from comparable sentences in (Barzilay and Lee, 2003) using multi-sentence alignment to discover "variable" and fixed

structures. Linguistic patterns were applied to huge amounts of non-annotated pre-classified texts in (Riloff, 1996) to bootstrap information extraction patterns. Similarly, semi-supervised or unsupervised methods have been used to learn question/answering patterns (Ravichandran and Hovy, 2002) or text schemas (Bunescu and Mooney, 2007). One current paradigm to learn from raw data is open information extraction (Downey et al., 2004; Banko, 2009), which without any prior knowledge aims at discovering all possible relations between pairs of entities occurring in text. Our work tries to learn the main concepts making up the template structure in domain summaries, similar to (Chambers and Jurafsky, 2011). However, we do not rely on any source of external knowledge (i.e. WordNet) to do so.

This paper presents an iterative-learning algorithm which is able to identify the key components of event summaries. We will show that the algorithm can induce template-like representations in various domains and languages. The rest of the paper is organized in the following way: In Section 2 we introduce the dataset we are using for our experiments and describe how we have prepated it for experimentation. Then, in Section 3 we provide an overview of our concept induction learnig algorithm while in Section 4 we explain how we have instantiated the algorithm for the experiments presented in this paper. Section 5 describe the experiments and results obtained and Section 6 discusses our approach comparing it with past research. Finally, in Section 7 we close the paper with conclusions and future work.

2 Data and Data Preparation

The dataset used for this study – part of the CONCISUS corpus (Saggion and Szasz, 2012) – consists of a set of 250 summaries in Spanish and English for three different domains: aviation accidents, rail accidents, and earthquakes. This dataset makes it possible to compare the performance of learning procedures across languages and domains. Based on commonsense, a human annotator developed an annotation schema per domain to describe in a template-like representation the essential elements (i.e., slots) of each event. For example, for the aviation accident domain these essential elements were: the date of the accident, the number of victims, the airline, the aircraft, the location of the accident, the flight number, the origin and destination of the flight, etc. The dataset was then annotated following the schema using the GATE annotation tool. The human annotations are used for evaluation of the concept discovery algorithm. Each document in the dataset was automatically annotated using tools for each language. We relied on basic processing steps to identify sentences, words and word-roots, parts of speech, noun-chunks, and named entities using the GATE system for English (Maynard et al., 2002) and TreeTagger for Spanish (Schmid, 1995).

Algorithm 1 Iterative Learning Algorithm: Main

```
1:  Given: C: Corpus of Summaries Annotated with Chunks
2:  Returns: LIST: A list of concepts discovered by the algorithm
3:  begin
4:  LIST ← ∅;
5:  while (EXIST_CONCEPTS_TO_LEARN) do
6:      CONCEPT ← LEAN_CONCEPT(C);
7:      if (not FILTER_CONCEPT(CONCEPT)) then
8:          LIST ← LIST ∪ CONCEPT;
9:      end if
10:     REMOVE_USED_CHUNKS(C);
11: end while
12: end
```

3 Learning Algorithm

The method is designed to learn the conceptual information in the summaries by extension (i.e., the set of strings that make up the concept in a given corpus) and by intension (i.e., an algorithm able to recognise the concept members in new documents in the domain) (Buitelaar and Magnini, 2005). Concept extensions identified by our method in the English summaries in the aviation domain are listed in Table 3. Each summary in the corpus can be seen as a sequence of strings and chunks as shown in Figure 1 (named entities and noun chunks are shown in boldface and they may overlap). The procedure to learn a concept in the corpus of summaries is given in pseudocode in Algorithm 2 which is repeatedly invoked by a main algorithm to learn all concepts (Algorithm 1).

The idea of the algorithm is rather simple, at each iteration a document is selected for learning, and from this document a single chunk (i.e., a noun chunk or a named entity) available for learning is selected as a seed example of a hypothetical concept (the concept is given a unique name at each itera-

Concept	Extension
1	Boeign 737-400; Boeign 777-200ER; Airbus 300; ...
2	August 16; December 20; February 12; ...
3	Colombia; Algiers; Brazil; Marseille; ...
4	102; 107; 145; 130; ...
5	Flight 243; Flight 1549; Flight 1907; ...
6	1988; 1994; 2001; ...

Table 1: Concepts Discovered in the Aviation Domain. They correspond (in order) to the type of aircraft, date of the incident, place of the accident, number of victims, flight number, and year of the accident.

tion). The document is annotated with this seed as a target concept and a classifier is trained using this document. The trained classifier is then applied to the rest of the documents to identify instances of the hypothetical concept. If the classifier is unsuccessful in identifying new instances, then the chunk used in the current iteration is discarded from the learning process, but if the classifier is successful and able to identify instances of the hypothetical concept, then the "best" annotated document is selected and added to the training set. The classifier is re-trained using the new added document and the process is repeated until no more instances can be identified. A hypothetical concept is kept only if there is enough support for it across the set of documents. The main procedure calls the basic algorithms a number of times while there are concepts to be learnt (or all chunks have been used). The stopping criteria is the number of concepts which could possibly be learnt, an estimation of which is the average number of chunks in a document.

4 Algorithm Instantiation

Experiments were carried out per domain and language to assess the suitability of the algorithm to the conceptual learning task. A number of points in Algorithm 2 need clarification: the selection of a document in line 4 of the algorithm can be carried out using different informed procedures; for the experiments described here we decided to select the document with more available hypotheses, i.e., the document with more chunks. For the selection of a

Algorithm 2 Iterative Learning Algorithm: Learn Concept

```
 1: Given: C: Corpus of summaries automatically annotated with named entities and chunks
 2: Returns: CONCEPT: A concept by extension and a trained algorithm to discover instances of the concept in text
 3: begin
 4: DOC ← SELECT_DOCUMENT(C);
 5: DOC ← ANNOTATE_WITH_TARGET(DOC);
 6: REST ← C \ {DOC};
 7: TRAINSET ← {DOC};
 8: CONTINUE ← true;
 9: while ((EXIST_DOCUMENTS_TO_LEARN) AND CONTINUE) do
10:    TRAIN(CLASSIFIER,TRAINSET);
11:    APPLY(CLASSIFIER,REST);
12:    if (DOCUMENT_LEANED(REST)) then
13:       BESTDOC ← SELECT_BEST(REST);
14:       TRAINSET ← TRAINSET ∪ {BESTDOC};
15:       REST ← REST \ {BESTDOC};
16:       CLEAN(REST);
17:    else
18:       CONTINUE ← false;
19:    end if
20: end while
21: CONCEPT ←< EXTENSION(TRAINSET); CLASSIFIER >;
22: return CONCEPT;
23: end
```

chunk to start the learning procedure in line 5 of the algorithm we select the next available chunk in text order. The classifier we used in line 10 of the algorithm is instantiated to Support Vector Machines (SVMs) which are distributed with the GATE system (Li et al., 2004). The features we use for representing the instance to be learnt are very superficial for these experiments: lemmas, parts-of-speech tags, orthography, and named entity types of the words surrounding the target concept to be learnt. The SVMs provide as output a class together with a probability which is essential to our method. We use this probability for selecting the best document in line 13 of the algorithm: the instance predicted with the highest probability is located and the document where this instance occurs is returned as "best document". In case no instances are learned (e.g., *else* in line 17), the iteration ends returning the extension learnt so far. Concerning Algorithm 1: in line 5 (the *while*) we use as stopping criteria for the maximum number of concepts to learn the average number of chunks in the corpus. In line 7, the FILTER_CONCEPT function evaluates the concept, keeping it only if two criteria are met: (i) there are not "too many" repetitions of a string in the discovered concept and (ii) the discovered concept covers

a reasonable number of documents. With criteria (i) we filter out a concept which contains repeated strings: a concept could be formed simply by grouping together all repeated phrases in the set of documents (i.e. "the earthquake" or "the accident" or "the plane"). While these phrases could be relevant in the target domain they do not constitute a key concept in our interpretation. Strings which are repeated in the concept extension are more like the "backbone structure" of the summaries in the domain. In our experiments both criteria are experimental variables and we vary them from 10% to 100% at 20% intervals. In Section 5 we will present results for the best configurations.

5 Experiments and Results

In order to evaluate the discovered concepts we have treated learning as information extraction. In order to evaluate them in this context we first need to map each learnt concept onto one of the human concepts. The mapping, which is based on the concept extension, is straightforward: a discovered concept is mapped onto the human concept with which it has a majority of string matches. Note that we match the discovered text offsets in the analysed documents and not only the identified strings. In order to evaluate the matching procedure we have used precision, recall, and f-score measures comparing the automatic concept with the human concept. Note that we use a lenient procedure – counting as correct strings those with a partial match. This is justified since discovering the exact boundaries of a concept instance is a very difficult task. Table 2 shows some examples of the human annotated instances and related discovered one. It can be appreciated that the learnt concepts have a reasonable match degree with the human annotated ones.

Table 3 gives information extraction results per domain and language for the best configuration of the algorithm. The best scores are generally obtained when coverage is set to 10% of the number of summaries, except for the learning of conceptual information in Spanish for the earthquake domain where the system performs better for 10% summary coverage. The parameter controlling string repetition in the concept extension should be kept small. The obtained results are quite satisfactory considering the small dataset and the limited use of linguistic resources during learning. These results compare favorably to cross-validation results obtained using supervised machine learning techniques (Saggion and Szasz, 2011). Learning from the earthquake domain appears to be more challenging given the more verbose characteristics of these texts. Even though space restricions prevent us from showing all evaluation results, in Table 4 we present detailed results for the two domains and languages. Note that the concepts listed constitute the slots of the induced domain template.

Annotated Instance	**Discovered Instance**
PMTair (Airline)	PMTair Flight
Boeing 777-200ER (TypeOfAircraft)	Boeing 777
the Margalla Hills northest of Islamabad (Place)	Margalla
transporte de mercancias (TypeOfTrain)	mercancias
29 abril 1997 (DateOfAccident)	29 abril

Table 2: Examples of Concept Extensions Partially Matched

Spanish			
Domain (% rep, % cov)	**Prec.**	**Rec.**	**F**
Aviation Accident (10%, 10%)	0.53	0.57	0.60
Rail Accident (10%, 10%)	0.66	0.67	0.66
Earthquake (10%, 30%)	0.41	0.30	0.35
English			
Domain	**Prec.**	**Rec.**	**F**
Aviation Accident (10%, 10%)	0.67	0.64	0.66
Rail Accident (30%, 10%)	0.52	0.33	0.44
Earthquake (10%, 10%)	0.40	0.19	0.26

Table 3: Performance in terms of Precision, Recall, and F-Score per Domain and Language. % rep and % cov are the repetition and coverage parameters used.

6 Discussion

Similar to active learning information extraction techniques (Ciravegna and Wilks, 2003), the concept discovery algorithm presented here is inspired by techniques like learning by reading, where unfamiliar expressions in one document can be "explained" by association to expressions in similar

English - Aviation Accidents			
Concept	**Precision**	**Recall**	**F-score**
Airline	0.90	0.90	0.90
DateOfAccident	0.90	0.93	0.92
FlightNumber	0.91	0.94	0.92
NumberOfVictims	0.41	0.30	0.35
Place	0.34	0.54	0.42
TypeOfAccident	0.42	0.76	0.54
TypeOfAircraft	0.73	0.75	0.74
Year	0.94	0.97	0.95
All	0.67	0.64	0.66
Spanish - Train Accidents			
Concept	**Precision**	**Recall**	**F-score**
DateOfAccident	1.00	1.00	1.00
NumberOfVictims	0.97	0.91	0.94
Place	0.43	0.76	0.55
Survivors	0.55	0.96	0.70
TypeOfAccident	0.74	0.63	0.68
TypeOfTrain	0.35	0.30	0.32
All	0.66	0.67	0.66
Spanish - Earthquakes			
Concept	**Precision**	**Recall**	**F-score**
Country	0.53	0.36	0.43
DateOfEarthquake	0.96	0.94	0.95
Fatalities	0.37	0.28	0.32
Magnitude	0.54	0.32	0.40
Region	0.16	0.56	0.25
All	0.35	0.35	0.35

Table 4: Learning Evaluation in the Train and Aviation Accident and Earthquake Domains (Spanish and English Dataset)

document contexts. However, and unlike active learning, human intervention is unnecessary in our approach. Although the algorithm achieves reasonably lenient performance, strict (hard) evaluation indicates that in each experimental condition performance drops when a strict match is required. This is expected given the difficulty of finding the right instance boundaries based only on automatic chunking information. For this reason, we intend to carry out additional experiments based on richer domain independent features from a syntactic parser. We have identified a number of reasons why some concept instances can not be correctly associated with their concepts. In the aviation domain, for example, numeric expressions constitute the extensions of different concepts including: number of victims, crew members, and number of survivors; it is a rather common feature in the aviation domain to include these different concepts together in one sentence, making their "separation" complicated. Same explanations apply to other tested domains: for example locations playing the role of origin and destination of a given train or airplace are also sometimes confused. Our work demonstrates the possibility of learning conceptual information in several domains and languages, while previous work (Chambers and Jurafsky, 2011) has addressed sets of related domains (e.g., MUC-4 templates) in English. Learning *full conceptualizations* from raw data is a daunting and difficult enterprise (Biemann, 2005). Here, we provide a short-cut by proposing a method able to learn the essential concepts of a domain by relying on summaries which are freely available on the Web. Our method is able to produce conceptualizations from a few documents in each domain and language unlike recent open domain information extraction which requires massive amount of texts for relation learning (Banko, 2009). Our algorithm has a reasonable computational complexity, unlike alignment-based or clustering-based approaches (Barzilay and Lee, 2003), which are computationally expensive.

7 Conclusions and Outlook

Domain adaptation is a time consuming and costly procedure calling for the development of algorithms and tools to facilitate its automation. In this paper we have presented a novel algorithm for learning information content in event summaries. The approach is fully unsupervised and based on the application of an iterative algorithm which grows a concept extension step-by-step. We have also proposed an instantiation of the algorithm and demonstrated its applicability to learning conceptual information in three different domains and two languages. We have obtained encouraging results, with the procedure able to model the main conceptual information in the summaries with lenient F-scores ranging from 0.25 to 0.66 F-scores depending on the language and domain. There are, however, a number of avenues that should be further explored such as the use of a richer document representation based on syntactic information and the development of additional procedures to improve instance boundary recognition.

Acknowledgements

We are grateful to the Advanced Research Fellowship RYC-2009-04291 from Programa Ramón y Cajal 2009, Ministerio de Economía y Competitividad, Secretaría de Estado de Investigación, Desarrollo e Innovación, Spain. We thank Biljana Drndarevic who helped proofreading the paper.

References

M. Banko. 2009. *Open Information Extraction for the Web*. Ph.D. thesis, University of Washington.

R. Barzilay and L. Lee. 2003. Learning to paraphrase: an unsupervised approach using multiple-sequence alignment. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 16–23, Stroudsburg, PA, USA. Association for Computational Linguistics.

C. Biemann. 2005. Ontology Learning from Text: A Survey of Methods. *LDV Forum*, 20(2):75–93.

P. Buitelaar and B. Magnini. 2005. Ontology learning from text: An overview. In *In Paul Buitelaar, P., Cimiano, P., Magnini B. (Eds.), Ontology Learning from Text: Methods, Applications and Evaluation*, pages 3–12. IOS Press.

R.C. Bunescu and R.J. Mooney. 2007. Learning to extract relations from the web using minimal supervision. In *ACL*.

N. Chambers and D. Jurafsky. 2009. Unsupervised learning of narrative schemas and their participants. In *ACL/AFNLP*, pages 602–610.

N. Chambers and D. Jurafsky. 2011. Template-Based Information Extraction without the Templates. In *ACL*, pages 976–986.

F. Ciravegna and Y. Wilks. 2003. Designing Adaptive Information Extraction for the Semantic Web in Amilcare. In S. Handschuh and S. Staab, editors, *Annotation for the Semantic Web*. IOS Press, Amsterdam.

G. DeJong. 1982. An Overview of the FRUMP System. In W.G. Lehnert and M.H. Ringle, editors, *Strategies for Natural Language Processing*, pages 149–176. Lawrence Erlbaum Associates, Publishers.

D. Downey, O. Etzioni, S. Soderland, and D. S. Weld. 2004. Learning Text Patterns for Web Information Extraction and Assessment. In *Proceedings of AAAI 2004 Workshop on Adaptive Text Extraction and Mining (ATEM'04)*.

Y. Li, K. Bontcheva, and H. Cunningham. 2004. An SVM Based Learning Algorithm for Information Extraction. Machine Learning Workshop, Sheffield.

P. Li, J. Jiang, and Y. Wang. 2010. Generating Templates of Entity Summaries with an Entity-Aspect Model and Pattern Mining. In *Proceedings of ACL*, Uppsala. ACL.

D. Maynard, V. Tablan, H. Cunningham, C. Ursu, H. Saggion, K. Bontcheva, and Y. Wilks. 2002. Architectural Elements of Language Engineering Robustness. *Journal of Natural Language Engineering – Special Issue on Robust Methods in Analysis of Natural Language Data*, 8(2/3):257–274.

Michael P. Oakes and Chris D. Paice. 2001. Term extraction for automatic abstracting. In D. Bourigault, C. Jacquemin, and M-C. L'Homme, editors, *Recent Advances in Computational Terminology*, volume 2 of *Natural Language Processing*, chapter 17, pages 353–370. John Benjamins Publishing Company.

D. Ravichandran and E. Hovy. 2002. Learning surface text patterns for a question answering system. In *ACL*, pages 41–47.

E. Riloff. 1996. Automatically generating extraction patterns from untagged text. *Proceedings of the Thirteenth Annual Conference on Artificial Intelligence*, pages 1044–1049.

H. Saggion and G. Lapalme. 2002. Generating Indicative-Informative Summaries with SumUM. *Computational Linguistics*, 28:497–526, December.

H. Saggion and S. Szasz. 2011. Multi-domain Cross-lingual Information Extraction from Clean and Noisy Texts. In *Proceedings of the 8th Brazilian Symposium in Information and Human Language Technology*, Cuiabá, Brazil. BCS.

H. Saggion and S. Szasz. 2012. The CONCISUS Corpus of Event Summaries. In *Proceedings of the 8th Language Resources and Evaluation Conference (LREC)*, Istanbul, Turkey. ELDA.

H. Saggion. 2011. Learning predicate insertion rules for document abstracting. In *Lecture Notes in Computer Science*, volume 6609, pages 301–312.

H. Schmid. 1995. Improvements in part-of-speech tagging with an application to german. In *In Proceedings of the ACL SIGDAT-Workshop*, pages 47–50.