Unsupervised Learning Summarization Templates 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
http://www.dtic.upf.edu/~hsaggion/

Abstract

We here present and compare two unsupervised approaches for inducing the main conceptual information in rather stereotypical summaries in two different languages. We evaluate the two approaches in two different information extraction settings: monolingual and cross-lingual information extraction. The extraction systems are trained on auto-annotated summaries (containing the induced concepts) and evaluated on human-annotated documents. Extraction results are promising, being close in performance to those achieved when the system is trained on human-annotated summaries.

1 Introduction

Information Extraction (Piskorski and Yangarber, 2013) and Automatic Text Summarization (Saggion and Poibeau, 2013) are two Natural Language Processing tasks which require domain and language adaptation. For over two decades (Riloff, 1993; Riloff, 1996) the natural language processing community has been interested in automatic or semi-automatic methods which could be used to port systems from one domain or task to another, aiming at reducing at least in part the cost associated with the creation of human annotated datasets. Automatic system adaptation can take different forms: if high quality human annotated data is available, then rule-based or statistical systems can be trained on this data (Brill, 1994), reducing the efforts of writing rules and handcrafting dictionaries. If high quality human annotated data is unavailable, a large non-annotated corpus and a bootstrapping procedure can be used to produce annotated data (Ciravegna and Wilks, 2003; Yangarber, 2003). Here, we concentrate on developing and evaluating automatic procedures to learn the *main concepts of a domain* and at the same time *auto-annotate texts* so that they become available for training information extraction or text summarization applications. However, it would be naive to think that in the current state of the art we would be able to learn all knowledge from text automatically (Poon and Domingos, 2010; Biemann, 2005; Buitelaar and Magnini, 2005). We therefore here concentrate on learning template-like representations from concise event summaries which should contain the key information of an event.

18 de julio de 1994$_{DateOfAttack}$. Un atentado contra la **sede de la Asociación Mutual Israelita Argentina**$_{Target}$ de **Buenos Aires**$_{PlaceOfAttack}$ causa la muerte de **86**$_{NumberOfVictims}$ personas.
(*18th July 1994. An attack against the headquarters of the Jewish Mutual Association in Buenos Aires, Argentina, kills 86 people.*)

Figure 1: Sample of Human Annotated Summary in Spanish

An example of the summaries we want to learn from is presented in Figure 1. It is a summary in the terrorist attack domain in Spanish. It has been

This work is partially supported by Ministerio de Economía y Competitividad, Secretaría de Estado de Investigación, Desarrollo e Innovación, Spain under project number TIN2012-38584-C06-03 and Advanced Research Fellowship RYC-2009-04291. We thank Biljana Drndarević for proofreading the paper.

Proceedings of NAACL-HLT 2013, pages 270–279,
Atlanta, Georgia, 9–14 June 2013.

manually annotated with concepts such as DateOfAttack, Target, PlaceOfAttack, and NumberOfVictims, which are key in the domain. Our task is to discover from this kind of summary what the concepts are and how to recognise them automatically. As will be shown in this paper and unlike current approaches (Chambers and Jurafsky, 2011; Leung et al., 2011), the methods to be presented here do not require parsing or semantic dictionaries to work or specification of the underlying number of concepts in the domain to be learn. The approach we take learns concepts in the set of domain summaries, relying on noun phrase contextual information. They are able to generate reasonable domain conceptualizations from relatively small datasets and in different languages.

The rest of the paper is structured as follows: In Section 2 we overview related work in the area of concept induction from text. Next, in Section 3 we describe the dataset used and how we have processed it while in Section 4 we outline the two unsupervised learning algorithms we compare in this paper for template induction from text. Then, in Section 5, we describe the experiments on template induction indicating how we have instantiated the algorithms and in Section 6 we explain how we have extrinsically evaluated the induction process. In Section 7 we discuss the obtained results and in Section 8 we summarize our findings and close the paper.

2 Related Work

A long standing issue in natural language processing is how to learn conceptualizations from text in automatic or semi-automatic ways. The availability of redundant data has been used, for example, to discover template-like representations (Barzilay and Lee, 2003) or sentence-level paraphrases which could be used for extraction or generation. Various approaches to concept learning use clustering techniques. (Leung et al., 2011) apply various clustering procedures to learn a small number of slots in three typical information extraction domains, using manually annotated data and fixing the number of concepts to be learnt. (Li et al., 2010) generate templates and extraction patterns for specific entity types (actors, companies, etc.). (Chambers and Jurafsky, 2011) learn the structure of MUC templates from raw data in English, an approach that needs both full parsing and semantic interpretation using WordNet (Fellbaum, 1998) in order to extract verb arguments and measure the similarity betweern verbs. In (Saggion, 2012) an iterative learning procedure is used to discover core domain conceptual information from short summaries in two languages. However, the obtained results were not assessed in a real information extraction scenario. There are approaches which do not need any human intervention or sophisticated text processing, but learn based on redundancy of the input dataset and some well grounded linguistic intuitions (Banko and Etzioni, 2008; Etzioni et al., 2004). Related to the work presented here are approaches that aim at generating short stereotypical summaries (DeJong, 1982; Paice and Jones, 1993; Ratnaparkhi, 2000; Saggion and Lapalme, 2002; Konstas and Lapata, 2012).

3 Dataset and Text Processing Steps

For the experiments reported here we rely on the CONCISUS corpus[1] (Saggion and Szasz, 2012) which is distributed free of charge. It is a corpus of Web summaries in Spanish and English in four different application domains: Aviation Accidents (32 English, 32 Spanish), Earthquakes (44 English, 56 Spanish), Train Accidents (36 English, 43 Spanish), and Terrorist Attacks (42 English, 53 Spanish). The dataset contains original and comparable summary pairs, automatic translations of Spanish summaries into English, automatic translation of English summaries into Spanish, and associated original full documents in Spanish and English for two of the domains (Aviation Accidents and Earthquakes). The dataset comes with human annotations representing the key information in each domain. In Table 1 we detail the concepts used in each of the domains. Note that not all concepts are represented in each of the summaries. Creation of such a dataset can take up to 500 hours for a human annotator, considering data collection, cleansing, and annotation proper. Only one human annotator and one curator were responsible for the annotation process.

[1] http://www.taln.upf.edu/pages/concisus/.

Aviation Accident	Airline, Cause, DateOfAccident, Destination, FlightNumber, NumberOfVictims, Origin, Passengers, Place, Survivors, Crew, TypeOfAccident, TypeOfAircraft, Year
Earthquake	City, Country, DateOfEarthquake, Depth, Duration, Epicentre, Fatalities, Homeless, Injured, Magnitude, OtherPlacesAffected, Province, Region, Survivors, TimeOfEarthquake
Terrorist Attack	City, Country, DateOfAccident, Fatalities, Injured, Target, Perpetrator, Place, NumberOfVictims, TypeOfAttack
Train Accident	Cause, DateOfAccident, Destination, NumberOfVictims, Origin, Passenger, Place, Survivors, TypeOfAccident, TypeOfTrain

Table 1: Conceptual Information in Summaries

3.1 Text Processing

In order to carry out experimentation we adopt the GATE infrastructure for document representation and annotation (Maynard et al., 2002). All documents in the dataset are processed with available natural language processors to compute shallow linguistic information. Documents in English are processed with the ANNIE system, a morphological analyzer, and a noun chunker, all three from GATE. The documents in Spanish are analyzed with TreeTagger (Schmid, 1995), a rule-base noun chunker, and an SVM-based named entity recognition and classification system.

4 Concept Induction Algorithms

Two algorithms are used to induce conceptual information in a domain from a set of textual summaries. The algorithms form concepts based on target strings (or chunks) in the set of summaries using token-level linguistic information. The chunks are represented with different features which are explained later in Section 5.1. One algorithm we use is based on *clustering*, while the other is based on *iterative learning*.

4.1 Clustering-based Induction

The procedure for learning conceptual information by clustering is straithforward: the chunks in the set of summaries are represented as instances considering both internal and sourrounding linguistic information. These instances are the input to a clustering procedure which returns a list of clusters each containing a set of chunks. We consider each cluster as a key concept in the set of domain summaries and the chunks in each cluster as the concept extension.

4.2 Iterative Induction

We use the iterative learning algorithm described in (Saggion, 2012) which learns from a set of summaries S, also annotated with target strings (e.g. chunks) and shallow linguistic information. In a nutshell the algorithm is as follows:

(1) Choose a document D from the set of summaries S and add it to a training set TRAIN. Set REST to $S -$ TRAIN.

(2) Choose an available target concept T from D, i.e. a target concept not tried before by the algorithm.

(3) Train a classifier on TRAIN to learn instances of the target concept using the available linguistic features; the classifier uses the linguistic information provided.

(4) Apply the classifier to REST (all summaries minus those in TRAIN) to annotate all instances of the target concept T.

(5) Select a document BEST in REST, where there is an instance of the concept recognised with the highest probability in the REST set.

(6) Remove BEST from REST and add BEST to the training set, remove all identified instances of T from REST, and go to step 3.

The algorithm is executed a number of times (see Section 5.1 for parametrization of the algorithms) to learn all concepts in the set of summaries, and at each iteration a single concept is formed. There are two circumstances when a concept being formed is discarded and their associated initial target concept removed from the learning process: one case is when there are not enough occurrences of the concept across a set of summaries; another case is when too many identical strings are proposed as instances for the concept in the set of summaries. This latter restriction is only valid if we consider sets of non-redundant documents, which is the case to which we restrict our experiments.

4.3 Text Chunks

Given that the algorithms presented above try to induce a concept from the chunks in the summaries,

we are interested in assessing how the type of chunk influences the learning process. Also, given that our objective is to test methods which learn with minimal human intervention, we are interested in investigating differences between the use of manual and automatic chunks. We therefore use the following chunk types in this work: *gold chunks (gold)* are the human produced annotations (as in Figure 1); *named entity chunks (ne)* are named entities computed by an off-the-shelf named entity recognizer; *noun chunks (nc)* are text chunks identified by rule-based off-the-shelf NP chunkers and finally, *wiki chunks (wiki)* are strings of text in the summaries which happen to be Wikipedia titles.

In order to automatically compute these chunk types, different levels of knowledge are needed. For example, NP chunks require syntactic information, while named entities and wiki chunks require some external form of knowledge, such as precompiled gazetteer lists or access to an encyclopædia or a semantic dictionary. Named entities and noun chunks are computed as described in Section 3, while wiki chunks are computed as follows: string n-grams $w_1 w_2 ... w_n$ are computed in each summary and strings $w_1\_w_2\_...\_w_n$ are checked against the Wikipedia on-line encyclopædia, if a hit occurs (i.e. if for an English n-gram the page en.wikipedia.org/wiki/$w_1 ... \_w_n$ exists or for a Spanish n-gram the page es.wikipedia.org/wiki/$w_1 ... \_w_n$ exists), the n-gram is annotated in the summary as a wiki chunk. Wiki chunks are cached to speed up the automatic annotation process.

	P	R	F
Spanish			
Terrorist Attack	0.47	0.10	0.17
Aviation Accident	0.52	0.08	0.14
Earthquake	0.24	0.06	0.10
Train Accident	0.59	0.15	0.24
English			
	P	R	F
Terrorist Attack	0.46	0.39	0.42
Aviation Accident	0.40	0.27	0.32
Earthquake	0.27	0.22	0.24
Train Accident	0.57	0.27	0.36

Table 2: Baseline Induction Performance

4.4 Mapping the Induced Concepts onto Human Concepts

For evaluation purposes, each induced concept is mapped onto one human concept applying the following procedure: let HC_i be the set of summary offsets where human concept i occurs, and let IC_i be the set of summary offsets where automatic concept i occurs, then the induced concept j is mapped onto concept k such that: $k = \arg\max_i(|\text{HC}_i \cap \text{IC}_j|)$, where $|X|$ is the size of set X. That is, the induced concept is mapped onto the label it gives it a best match. As an example, one induced concept in the terrorist attack domain containing the following string instances: *two bombs*, *car bomb*, *pair of bombs*, *10 coordinated shooting and bombing*, *two car bombs*, *suicide bomb*, *the attack*, *guerrilla warfare*, *the coca-growing regions*, etc. This induced concept is mapped onto the TypeOfAttack human concept in that domain.

4.5 Baseline Concept Induction

A baseline induction mechanism is designed for comparison with the two learning procedures proposed here. It is based on the mapping of named entity chunks onto concepts in a straightforward way: each named entity type is considered a different concept and therefore mapped onto human concepts as in Section 4.4. For example, in the terrorist attack domain, *Organizatio*n named entity type is mapped by this procedure onto the human concept Target (i.e. churches, government buildings, etc., are common targets in terrorist attacks) while in the Aviation Accident domain the *Organization* named entity type is mapped onto TypeOfAircraft (i.e. Boeing, Airbus, etc. are names of organizations).

5 Experimental Setting and Results of the Induction Process

In this section we detail the different parameters used by the algorithms and report the performance of the induction process with different inputs.

5.1 Settings

The features used by the induction procedure are extracted from the text tokens. We extract the POS tag, root, and string of each token. The clustering-based algorithm uses a standard Expectation Maximization

Spanish						
	Iterative			Clustering		
	P	R	F	P	R	F
Terrorist Attack	0.25	0.59	0.35	0.59	0.59	0.59†
Aviation Accident	0.50	0.62	0.55	0.66	0.66	0.66†
Earthquake	0.34	0.51	0.41	0.56	0.53	0.55†
Train Accident	0.41	0.69	0.52	0.58	0.58	0.58
English						
	Iterative			Clustering		
	P	R	F	P	R	F
Terrorist Attack	0.23	0.39	0.29	0.50	0.50	0.50†
Aviation Accident	0.57	0.68	0.62	0.79	0.79	0.79†
Earthquake	0.26	0.53	0.34	0.39	0.39	0.39
Train Accident	0.50	0.59	0.54	0.61	0.61	0.61†

Table 3: Conceptual induction (Spanish and English) Using Gold Chunks for Learning

implementation from the Weka machine learning library (Witten and Frank, 1999). We instruct the algorithm to decide on the number of clusters based on the data, instead of setting the number of clusters by hand. The instances to cluster are representations of the input chunks; these representations contain the internal features of the chunks, as well as the information of 5 tokens to the left of the beginning of the chunk and 5 tokens to the right of the end of the chunk. The transformation from GATE documents into arff Weka files and the mapping from Weka onto the GATE documents, is carried out using specific programs. The classification algorithm used for the iterative learning process is an SVM classifier distributed with the GATE system and tuned to perform chunk learning using the same features as the clustering procedure (Li et al., 2004). This classifier outputs a probability which we use for selecting the best document at step (5) of the iterative procedure. The document selected to start the process is the one with more target strings, and the target string chosen is the next available in textual order. The iterative learning procedure is set to stop when the number of concepts induced reaches the average number of chunks in the corpus. Induced concepts not covering at least 10% of the number of documents are discarded, as are concepts with strings repeated at least 10% of the concept extension.

5.2 Experiments and Results

We carry out a number of experiments per domain where we run the algorithms using as input the summaries annotated with a different chunk type each time. After each experiment all concepts induced are

Terrorist Attacks						
	Iterative			Clustering		
	P	R	F	P	R	F
nc	0.22	0.53	0.31†	0.15	0.51	0.23
ne	0.27	0.14	0.18	0.12	0.42	0.18
wiki	0.15	0.26	0.19	0.22	0.18	0.20
all	0.25	0.53	0.34†	0.12	0.51	0.20
Aviation Accidents						
	Iterative			Clustering		
	P	R	F	P	R	F
nc	0.30	0.50	0.38†	0.21	0.51	0.30
ne	0.84	0.07	0.14	0.57	0.07	0.13
wiki	0.29	0.28	0.28†	0.27	0.17	0.21
all	0.39	0.62	0.48†	0.16	0.31	0.21
Earthquakes						
	Iterative			Clustering		
	P	R	F	P	R	F
nc	0.29	0.42	0.34†	0.14	0.42	0.21
ne	0.20	0.19	0.20†	0.38	0.02	0.05
wiki	0.16	0.16	0.16	0.24	0.11	0.15
all	0.28	0.50	0.36†	0.12	0.46	0.19
Train Accidents						
	Iterative			Clustering		
	P	R	F	P	R	F
nc	0.36	0.66	0.47†	0.23	0.51	0.32
ne	0.33	0.66	0.44†	0.65	0.12	0.20
wiki	0.25	0.25	0.25	0.51	0.13	0.21
all	0.33	0.62	0.44†	0.16	0.50	0.24

Table 4: Comparison of conceptual induction in Spanish

mapped onto the human concepts (see Section 4.4) producing auto-annotated summaries. The automatic annotations are then compared with the gold annotations, and precision, recall, and f-score figures are computed to observe the performance of the two algorithms, the baseline, and the effect of type of chunk on the learning process.

In Table 2 we report baseline performance on the entire dataset. As can be appreciated by the obtained numbers, directly mapping named entity types onto concepts does not provide a very good performance, especially for Spanish; we expected the learning procedures to produce better results. In Table 3 we present the results of inducing concepts from the gold chunks by the two algorithms. In almost all cases, using gold chunks improves over the baseline procedure, except for the Terrorist Attack domain in English, where the iterative learning procedure underperforms the baseline. In all tested domains, the clustering-based induction procedure has a very competitive performance. A t-test is run to verify differences in performance between the two systems in terms of f-score. In all tested domains in Spanish, except the Train Accident domain, there are sta-

Terrorist Attacks

	Iterative			Clustering		
	P	R	F	P	R	F
nc	0.43	0.50	0.46†	0.23	0.42	0.30
ne	0.28	0.44	0.34	0.42	0.29	0.34
wiki	0.24	0.33	0.28†	0.15	0.25	0.19
all	0.31	0.49	0.38†	0.09	0.39	0.15

Aviation Accidents

	Iterative			Clustering		
	P	R	F	P	R	F
nc	0.48	0.31	0.38	0.33	0.34	0.34
ne	0.53	0.38	0.44†	0.63	0.27	0.38
wiki	0.31	0.44	0.36†	0.28	0.37	0.32
all	0.50	0.67	0.58†	0.15	0.47	0.23

Earthquakes

	Iterative			Clustering		
	P	R	F	P	R	F
nc	0.29	0.48	0.36†	0.06	0.40	0.10
ne	0.28	0.34	0.30	0.30	0.25	0.28
wiki	0.21	0.30	0.25†	0.16	0.23	0.19
all	0.31	0.44	0.37†	0.08	0.40	0.13

Train Accidents

	Iterative			Clustering		
	P	R	F	P	R	F
nc	0.45	0.54	0.49†	0.32	0.50	0.39
ne	0.47	0.29	0.36	0.58	0.27	0.36
wiki	0.51	0.32	0.39†	0.30	0.29	0.29
all	0.50	0.58	0.53†	0.16	0.49	0.24

Table 5: Comparison of conceptual induction in English

	Spanish		
	P	R	F
Aviation Accident	0.83	0.60	0.70
Earthquake	0.61	0.48	0.53
Train Accident	0.77	0.54	0.64
	English		
	P	R	F
Aviation Accident	0.88	0.38	0.53
Earthquake	0.86	0.56	0.68
Train Accident	0.84	0.43	0.57

Table 6: Cross-lingual Information Extraction. System Trained with Gold Summaries.

Aviation Accidents

	Iterative			Clustering		
	P	R	F	P	R	F
gold	0.85	0.52	0.65†	0.84	0.41	0.55
all	0.88	0.49	0.63†	0.87	0.19	0.32
nc	0.87	0.46	0.60	0.88	0.46	0.60

Earthquakes

	Iterative			Clustering		
	P	R	F	P	R	F
gold	0.65	0.41	0.50†	0.66	0.31	0.43
all	0.64	0.36	0.46	0.62	0.40	0.49
nc	0.63	0.33	0.43	0.67	0.38	0.49

Train Accidents

	Iterative			Clustering		
	P	R	F	P	R	F
gold	0.81	0.54	0.65	0.82	0.52	0.64
all	0.81	0.52	0.64†	0.72	0.31	0.43
nc	0.79	0.54	0.64†	0.79	0.42	0.55

Table 7: Cross-lingual Information Extraction Results in Spanish Translations. System trained with auto-annotated summaries in Spanish.

Aviation Accidents

	Iterative			Clustering		
	P	R	F	P	R	F
gold	0.87	0.35	0.50	0.87	0.37	0.52
all	0.87	0.37	0.52†	0.82	0.18	0.29
nc	0.90	0.21	0.34†	0.90	0.17	0.29

Earthquakes

	Iterative			Clustering		
	P	R	F	P	R	F
gold	0.87	0.53	0.66†	0.87	0.36	0.51
all	0.88	0.51	0.64†	0.87	0.30	0.45
nc	0.88	0.51	0.65†	0.93	0.43	0.59

Train Accidents

	Iterative			Clustering		
	P	R	F	P	R	F
gold	0.82	0.30	0.44	0.87	0.32	0.47
all	0.84	0.39	0.53†	0.91	0.24	0.38
nc	0.89	0.36	0.51†	0.46	0.25	0.32

Table 8: Cross-lingual Information Extraction Results in English Translations. System trained with auto-annotated summaries in English.

tistically significant differences between the clustering procedure and the iterative learning procedure ($p = 0.01$). In all tested domains in English, except for the Earthquake domain, there are statistically significant differences between the performance of clustering and iterative learning ($p = 0.01$).

Now we turn to the results of both algorithms when automatic chunks are used, that is, when no human annotation is provided to the learners. Results are reported in Tables 4 (Spanish) and 5 (English). The results are presented by the chunk type used during the learning procedure. In addition to the chunk types specified above, we include a type **all**, which represents the use of all automatically computed chunks (i.e. **nc**, **ne**, **wiki**). We observe that, in general, when presented with automatic chuks, the iterative learning procedure is able to induce concepts with a better f-score than the clustering-based algorithm. A t-test is run to verify differences between the two induction procedures within each chunk condition (differences shown with a † in the tables). In 11 out of 16 cases in Spanish and in 12 out of 16 cases in English, statistically significant differences are observed. In three out of four domains the combination of automatic chunks outperforms the use of individual chunk types. Generally, named entity chunks and wiki chunks have the lowest performance. This is

	Spanish		
	P	R	F
Aviation Accident	0.56	0.47	0.51
Earthquake	0.64	0.41	0.50
	English		
	P	R	F
Aviation Accident	0.61	0.35	0.44
Earthquake	0.78	0.41	0.54

Table 9: Extraction from Full Documents. System Trained on Gold Summaries.

	Aviation Accidents					
	Iterative			**Clustering**		
	P	R	F	P	R	F
gold	0.55	0.37	0.44	0.54	0.31	0.39
all	0.55	0.36	0.43†	0.69	0.17	0.27
nc	0.45	0.22	0.30†	0.52	0.26	0.35
	Earthquake					
	Iterative			**Clustering**		
	P	R	F	P	R	F
gold	0.62	0.31	0.41†	0.63	0.22	0.33
all	0.61	0.26	0.37	0.63	0.31	0.41†
nc	0.60	0.24	0.35	0.70	0.28	0.40†

Table 10: Full-text Information Extraction Results in Spanish. System trained with auto-annotated summaries in Spanish.

not an unexpected result since named entities, for example, cover much fewer strings which may form part of a concept extension. Additionally, off-the-shelf entity recogizers only identify a limited number of entity types.

6 Information Extraction Evaluation Framework

The numbers above are interesting because they provide intrinsic evaluation of the concept induction procedure, but they do not tell us much about their usability. Therefore, and in order to better assess the value of the discovered concepts, we decided to carry out two extrinsic evaluations using an information extraction task. Once the conceps are induced and, as a result, the summaries are auto-annotated with domain specific concepts, we decide to train an off-the-shelf SVM token classification procedure and apply it to unseen human annotated documents. The SVM classifier uses the same linguistic information as the induction procedures: token level information and a window size of 5 around each token to be classified.

	Aviation Accidents					
	Iterative			**Clustering**		
	P	R	F	P	R	F
gold	0.60	0.28	0.39	0.62	0.31	0.41†
all	0.62	0.30	0.41†	0.54	0.14	0.23
nc	0.53	0.15	0.23†	0.46	0.10	0.16
	Earthquake					
	Iterative			**Clustering**		
	P	R	F	P	R	F
gold	0.70	0.35	0.47†	0.72	0.32	0.44
all	0.74	0.37	0.49	0.70	0.22	0.34
nc	0.73	0.36	0.48†	0.73	0.30	0.42

Table 11: Full-text Information Extraction Results in English. System trained with auto-annotated summaries in English.

6.1 Extraction from Automatic Translations

The first task we carry out is cross-lingual information extraction where the input documents are automatic translations of summaries in Spanish and English[2]. Note that the expriment is performed in three domains for which such translations are manually annotated. We first run an experiment to assess the extraction performance of the SVM when trained on human annotated data. Results of the experiment are reported in Table 6 and they should be taken as an upperbound of the performance of a system trained on auto-annotated summaries. We then train the SVM on the different auto-annotated datasets, but note that due to space restrictions, we here only report the three most revealing experiments per language: concepts induced with gold chunks, noun chunks, and all automatic chunks. Results are reported in Table 7 (Spanish) and in Table 8 (English). In most cases the SVM trained with auto-annotated summaries produced by the iterative learning procedure outperforms the clustering-based method with statistically significant differences († shown in the tables) ($p = 0.01$).

6.2 Extraction from Full Documents

The second and the last evaluation consists in the application of the SVM extraction system to full documents. In this case, the experiment can be run only in two domains for which full documents have been provided and manually annotated. We first test the performance of the system when trained on human annotated summaries and present the results in Table 9. Results of the experiments when the system is trained on auto-annotated datasets are shown in

[2]The translations were produced by Google translator.

Tables 10 (Spanish) and 11 (English). Results are lower than when training on clean human annotated summaries. It is unclear which approach is more competitive when training with auto-annotated summaries. What is clear is that the performance of the iterative learning algorithm when training with concepts induced from gold chunks is not statistically different (according to a t-test and $p = 0.01$) from the performance of the algorithm when training with concepts induced from automatically computed chunks. We consider this to be a positive outcome of the experiments.

7 Discussion

The two methods presented here are able to produce partial domain conceptualizations from a relatively small set of domain summaries[3]. We have found that the clustering-based procedure is very competitive when presented with gold chunks. On the other hand, the iterative learning procedure performs very well when presented with automatic chunks in all tested domains and the two languages. We have also found that the performance of the iterative induction system is not much affected by the use of automatically computed chunks. We have run a t-test to verify the differences in induction performance when learning with gold and automatic chunks (*all* condition) and have found statistically significant differences in only one domain out of four in Spanish (Terrorist Attack) and in two domains out of four in English (Aviation Accident and Train Accident) ($p = 0.01$). The applicability of the induction process, that is, if the auto-annotated data could be used for specific tasks, has been tested in two information extraction experiments. In a cross-lingual information extraction setting (Riloff et al., 2002; Saggion and Szasz, 2011) we have observed that a system trained on automatically computed chunks has a performance close to one trained on concepts induced from gold chunks. No statistically significant differences exist ($p = 0.01$) between the use of automatic chunks and gold chunks, except for the Train Accident domain in English, where the system trained on fully automatically annotated summaries has a better performance. In a full document information extraction task, although the best system trained on auto-annotated summaries in Spanish has a big difference with respect to a system trained on human-annotated summaries, in English the differences are slight. We belive that this is due to the differences in performance between the underlying text processing components. Our methods work by grouping together sets of chunks, unlike (Chambers and Jurafsky, 2011), whose approach is centered around verb arguments and clustering, and relies on the availability of considerable amounts of data. Ontology learning approaches such as OntoUSP (Poon and Domingos, 2010) are also clustering-based but focus on learning is-a relations only. Unlike (Leung et al., 2011) whose approach is based on gold-standard humman annotations, we here test the performance of the induction process using automatically computed candidate strings, and we additionally learn the number of concepts automatically.

[3]Depending on the language and domain, between 50% and 77% of all concepts are generated.

8 Conclusions and Future Work

In this paper we have concentrated on the problem of knowledge induction from text summaries. The approaches we have presented are fully unsupervised and are able to produce reasonable conceptualizations (close to human concepts) without relying on annotated data. Unlike previous work, our approach does not require full syntactic parsing or a semantic dictionary. In fact, it only requires a process of text chunking and named entity recognition, which we have carefully assessed here. We believe our work contributes with a viable methodology to induce conceptual information from texts, and at the same time with an auto-annotation mechanism which could be used to train information extraction systems. Since our procedure requires very little linguistic information, we believe it can be successfully applied to a number of languages. We also believe that there is much work to be carried out and that induction from summaries should be complemented with a process that explores full event reports, in order to reinforce some induced concepts, discard others, and discover additional ones.

References

Michele Banko and Oren Etzioni. 2008. The tradeoffs between open and traditional relation extraction. In

Proceedings of ACL-08, pages 28–36. Association for Computational Linguistics, June.

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.

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

E. Brill. 1994. Some Advances in Transformation-Based Part of Speech Tagging. In *Proceedings of the Twelfth National Conference on AI (AAAI-94)*, Seattle, Washington.

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.

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

Fabio Ciravegna and Yorick Wilks. 2003. Designing adaptive information extraction for the semantic web in amilcare. In *Annotation for the Semantic Web, Frontiers in Artificial Intelligence and Applications. IOS*. Press.

Gerald 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.

O. Etzioni, M. Cafarella, D. Downey, A. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. 2004. Methods for Domain-Indepedent Information Extraction from the Web: An Experimental Comparison. In *Proceedings of AAAI-2004*.

Christiane Fellbaum, editor. 1998. *WordNet - An Electronic Lexical Database*. MIT Press.

I. Konstas and M. Lapata. 2012. Concept-to-text generation via discriminative reranking. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 369–378, Jeju Island, Korea, July. Association for Computational Linguistics.

Cane Wing-ki Leung, Jing Jiang, Kian Ming A. Chai, Hai Leong Chieu, and Loo-Nin Teow. 2011. Unsupervised information extraction with distributional prior knowledge. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 814–824, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

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.

Chris D. Paice and Paul A. Jones. 1993. The Identification of Important Concepts in Highly Structured Technical Papers. In R. Korfhage, E. Rasmussen, and P. Willett, editors, *Proc. of the 16th ACM-SIGIR Conference*, pages 69–78.

J. Piskorski and R. Yangarber. 2013. Information extraction: Past, present and future. In Thierry Poibeau, Horacio Saggion, Jakub Piskorski, and Roman Yangarber, editors, *Multi-source, Multilingual Information Extraction and Summarization*, Theory and Applications of Natural Language Processing, pages 23–49. Springer Berlin Heidelberg.

H. Poon and P. Domingos. 2010. Unsupervised ontology induction from text. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 296–305, Stroudsburg, PA, USA. Association for Computational Linguistics.

Adwait Ratnaparkhi. 2000. Trainable methods for surface natural language generation. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, NAACL 2000, pages 194–201, Stroudsburg, PA, USA. Association for Computational Linguistics.

E. Riloff, C. Schafer, and D. Yarowsky. 2002. Inducing information extraction systems for new languages via cross-language projection. In *Proceedings of the 19th international conference on Computational linguistics*, pages 1–7, Morristown, NJ, USA. Association for Computational Linguistics.

E. Riloff. 1993. Automatically constructing a dictionary for information extraction tasks. *Proceedings of the Eleventh Annual Conference on Artificial Intelligence*, pages 811–816.

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*.

H. Saggion and T. Poibeau. 2013. Automatic text summarization: Past, present and future. In Thierry Poibeau, Horacio Saggion, Jakub Piskorski, and Roman Yangarber, editors, *Multi-source, Multilingual Information Extraction and Summarization*, Theory and Applications of Natural Language Processing, pages 3–21. Springer Berlin Heidelberg.

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. 2012. Unsupervised content discovery from concise summaries. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, AKBC-WEKEX '12, pages 13–18, Stroudsburg, PA, USA. Association for Computational Linguistics.

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.

I. H. Witten and E. Frank. 1999. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann.

R. Yangarber. 2003. Counter-Training in Discovery of Semantic Patterns. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL'03)*.