Indirect Supervised Learning of Content Selection Logic

Pablo A. Duboue

Columbia University Dept. of Computer Science
New York, NY, 10025, USA
pablo@cs.columbia.edu
http://www.cs.columbia.edu/~pablo/

Abstract. I investigate the automatic acquisition of Content Selection (CS) rules; a desirable goal, as the CS problem is quite domain dependent. My learning uses a loosely aligned Text-Data corpus, a resource increasingly popular in learning for NLG because they are readily available and do not require expensive hand labelling. However, they only provide indirect information about the selected or not selected status of each semantic datum. Indirect Supervised Learning is my proposed solution to this problem, a solution common to other learning from loosely aligned Text-Data corpora problems in NLG. It has two steps; in the first step, the loosely aligned Text-Data corpus is transformed into a dataset with classification labels. In the second step, supervised learning machinery acquires the CS rules from this dataset. I evaluate the approach by comparing the output of my system with the information selected by human authors in unseen texts.

Keywords. Machine Learning, Content Selection, Statistical Methods, Aligned Text-Data Corpora.

1 Introduction

Content Selection (CS), the problem of choosing the right information to communicate in the output, is in general a highly domain dependent task; new CS rules must be developed for each new domain. This is typically a tedious task, as a realistic knowledge base contains large amounts of data selected.

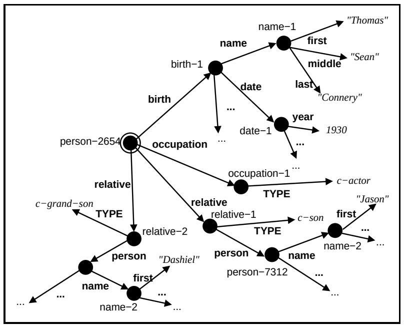
I am interested in automatically acquiring a set of CS rules in a biographical description domain. I want to learn these rules from a training dataset consisting of input data and classification labels (selected or not-selected). These rules decide whether or not to include a piece of information based solely on the semantics of the data (e.g., the relation of the information to other data in the input). However, that would require expensive hand-labelling. Instead, I turn to a loosely aligned Text (Fig. 1 (a)) and Data (Fig. 1 (b)) corpus. Loosely aligned Text-Data corpora are increasingly popular in learning for NLG because they are readily available[1] and do not require expensive hand labelling. However,

[1] Examples of domains with available text data corpora include Biology (e.g., the biological KB and species descriptions); Geography (e.g., CIA fact-book and country

they only provide indirect information about the selected or not selected status of each semantic datum. Indirect Supervised Learning[2] is my proposed solution to this problem, arguably a solution common to other learning from Text-Data corpora problems in NLG. It has two steps; in the first step, the Text-Data corpus is transformed into a training dataset of selected or not-selected classification labels. In the second step, CS rules are learned in a supervised way, from the training dataset constructed in the previous step.

Actor, born Thomas Connery on August 25, 1930, in Fountainbridge, Edinburgh, Scotland, the son of a truck driver and charwoman. He has a brother, Neil, born in 1938. Connery dropped out of school at age fifteen to join the British Navy. Connery is best known for his portrayal of the suave, sophisticated British spy, James Bond . . .

(a) Text.



(b) Data.

Fig. 1. Loosely aligned Text-Data corpus.

The output of my machine learning system (Fig. 4) will select data that can be further filtered and ordered by later stages in the generation pipeline (e.g., see the spreading activation algorithm used in ILEX [1]). Because my knowledge representation allows for cycles, the actual knowledge representation is as a graph: each frame is a node on it and there are edges labelled with the attribute names joining the different nodes. Atomic values are also represented as special nodes. In this way, my CS rules contain decision logic only for these special atomic nodes. In this paper, expressions such as "a piece of data to be selected" are formalized as one of these atomic nodes in the graph.

In previous work [2], we presented a version of our system for learning CS rules from a Text-Data corpus. The type of rules mined were very general, in the form of Select-All/Select-None type of rules (e.g., select all relative first names or do not select any of them). In this work, I defined a rule language and perform a Genetic Search over the possible rules on that language, picking the one that better explains a training dataset, built from the Text-Data corpus. This new, improved approach let me augment the quality of the learned rules,

descriptions); Financial Market (e.g., Stock data and market reports); and Entertainment (e.g., Role Playing character sheets and character descriptions).

[2] Indirect Supervised Learning, where the system learns from indirect, teacher provided examples is not to be confused with Semi-supervised learning. The latter is a bootstrapping method while the former is related to reinforcement learning.

making them closer to the type of rules NLG practitioners write to solve CS tasks.

This paper is organized as follows: in the next Section, my domain (together with the loosely aligned Text-Data corpus internals) is introduced. My methods are presented in Sec. 3 followed by the experimental results. Related work and conclusions are summarized at the end of the paper.

2 Domain: Biographical Descriptions

My technique applies to descriptive texts which realize a single, purely informative, communicative goal, as opposed to cases where more knowledge about speaker intentions are needed. In particular, I focus on biography generation, an exciting field that has attracted practitioners of NLG in the past [3–6].

I have gathered a loosely aligned Text-Data corpus in this domain. The Data part contains information related to the person the biography talks about (and that the system will use to generate that type of biography). However, not all that information will necessarily appear in the biography. That is, the Data part is not the semantics of the *target text* but the larger set of all things that could possibly be said about the person in question.

To collect the Data, I crawled 1,100 HTML pages containing celebrity fact-sheets from the *E! Online* website.[3] I then proceeded to transform the information in the pages to a frame-based knowledge representation (the final corpus contains aprox. 50K frames).

The text part was mined from two different web-sites, `biography.com`, containing typical biographies, with an average of 450 words each; and `imdb.com`, the Internet movie database, 250-word average length biographies.

3 Methods

I present here my two step learning process. In the first step (Sec. 3.1), I use the Text-Data corpus to infer the classification labels, selected or not-selected, in the data side (that constitutes the training instances for the supervised learning). In the second step (Sec. 3.2), I train a supervised classifier for the CS task.

3.1 Dataset Construction

The data selected for inclusion in the text can be divided into two classes: data that appears verbatim in the text and data that is somehow verbalized. To deal with the data copied verbatim, I search for it in the text, in a process I call *exact matching,* described next. Finally, the rest of the selected data is verbalized (e.g., `c-comedian` ⇒ *"comedian"*). I introduce a *statistical selection* process to target these harder cases.

[3] This process is not required to build a loosely aligned Text-Data corpus, I crawled fact-sheets to provide structured knowledge while I was building the Knowledge Component (that uses information extraction) for the ProGenIE generator [7].

Exact Matching. Exact matching aims to identify pieces from the input that are copied verbatim to the output. When I found a piece of data appearing in the text, I conclude that the data must be selected for inclusion in the text. The output of this process is a set of labels (selected or not-selected) for each datum in the input. This training dataset is to be used for supervised learning of CS rules.

The overall matching operation is prone to two type of errors: omission (if the atomic value is symbolic in nature, e.g., `c-comedian` appears in the text as *"he performed comedy for several years"* or a different verbalization, e.g., *"MA"* instead of *"Maryland"*) and over-generation (if there are several values for it, e.g., *"Smith"* appears in the text and it is simultaneously the name of the father and a brother). The latter errors were addressed with a post-processing step based on automatically mined defeasible rules [8], that I cannot detail here by lack of space. More interestingly, the former type of errors are addressed with the statistical selection technique described below.

Statistical Selection. My statistical selection algorithm aims to identify variations on the data that produce variations on the text side. For example, all the comedians (a semantic distinction induced from the data side) may have their biographies with words such as *"comedy," "stand-up"* and the like occurring on them more often than in the rest of the biographies. To this end, I first generate meaningful variations on the data side and then apply statistical filters to the resulting partition of documents on the text side.

In previous work [2], we employed clustering on the atomic values to induce the variations in the data side. I extend that work by replacing the clustering step with an enumeration process. To see the reasons behind this extension, consider the following example: in [2], we clustered all people who were born in November and compared their biographies with the ones of the people born in December. That approach turned up to do well. But now consider that clustering on the atomic values (November, December, etc.) will not necessarily be so informative in cases such as first names. Putting together all the relatives with named *"John"* against the relatives named *"Peter"* is not a semantically meaningful distinction on the data-side. In this extension, therefore, I was interested in enumerating all rules that partition the input semantic data into two sets of a minimum size (or *minimum support*). I employed then *complete level-wise search*[4] [9] to obtain them.

These rules belong to the same language of the final rules mined by the system (Fig. 4). They are used to discriminate the full set of knowledge representations. If any node on a knowledge representation is selected when the rule is applied to it (that is, the rule returns **true**), then the whole knowledge representation is

[4] Complete level-wise search is employed to search for all instances verifying a certain property. Each instance has an associated complexity (its *level*) and the property must be anti-monotonic: the higher the level, the smaller the number of instances that verify the property. By enumerating the instances in a level-wise manner, all instances that verify the property are guaranteed to be found in an efficient fashion.

selected for the cluster. The rules are enumerated as follows: for each data-path, I record all possible atomic values and measure its support. I then perform a number of passes (normally between 2 or 3), joining atomic values (in the form of `IN(...)` rules). At all times, I keep only the ones that have enough *support*.

After the atomic rules have been enumerated, I grow paths on the graph, also by doing a number of passes (normally between 3 to 5). In each pass, a working set goes through a number of steps. In the forward step, I take all the nodes reachable from the nodes in the data class through a path in the working set and record which labels **depart** from them. The extension of the paths using these edges form a candidate set that I then filter for paths without enough support. In the backward step, I build now a candidate set by taking the paths in the working set and going back to their parents. Again, I filter the paths by support. The new paths constitute the working set for the next round. Finally, I combine all the paths that had enough support in this enumeration process with the atomic values from the previous step to form `TRAVERSE` rules. The paths alone are also employed to form `TRAVERSE-EQ` rules. The resulting rule set is again checked for having enough support and for rules selecting the same set of instances (that are deemed as synonyms and all but one are discarded).

Finally, I create advanced rules with `AND` and `OR`. These rules do not follow level-wise behavior, and I thus only select a subset of them, up to some complexity level. I combine rules from a working set a number of times (normally 2 or 3).

With this rule-induced partition of the training data I then look at the text side, looking for difference on the language employed in each partition. In previous work, I employed cross-entropy of the language models induced by the partition. This solution is satisfactory in pin-pointing a global variation on the overall texts, but I wanted to obtain further information about the difference between the partitions.[5] In this work, I investigated statistical tests on the counts for each word on either partition. For example, if the partition is induced by putting aside all comedians and writers in the input data, I then want to conclude that their associated biographies contain the words *"comedian," "wrote,"* and *"producer"* more than expected by chance on this corpus (Fig. 2).

Finally, I am left with two training datasets containing classification labels for each piece of input data: one dataset produced by the exact matching process and another dataset produced by the statistical selection process. I need to combine them into a sole dataset for the supervised learning. I investigated two ways to combine the data: full union and selected union. In the full union, I just performed a logical OR of the information contained in both datasets. However, this degraded the results, because the statistical selection data is much noisier than the exact match data. Therefore, I introduced selected union, where the datasets are pre-screened and certain types of input data are marked as always exact matched (if the number of exact matched instances is higher than a

[5] Cross-entropy is a feasible solution to this problem, but its results are difficult to interpret. Furthermore, the extracted words of the new statistical tests can be later on used to obtain training material for learning ordering (not only selection).

for relative #TYPE rule `IN("c-sister-in-law", "c-sister")`; words: *sister, often*

for occupation #TYPE –

- rule `EQ("c-occupation-writer")`; words: *during, which, own,* ***writer,*** *from, years, also, success*
- rule `IN("c-occupation-model", "c-occupation-comedian")`; words: ***modeling,*** *appearances, turning*
- rule `TRAVERSE("../#TYPE",f, IN("c-occupation-comedian", "c-occupation-writer"))`; words: *producer, wrote, comedian, which, own, from, order, success*

Fig. 2. Extracted Words.

percentage threshold on their total number). This ensures a balance between the information coming from the exact match data set and the statistical selection dataset, augmenting the signal-to-noise ratio in the final dataset.

3.2 Supervised Learning

From the Data Construction step, I have a dataset consisting of classification labels (selected, not-selected) for each piece of input data. I want to learn that mapping (from datum to classification label) and capture that knowledge in the form of CS rules. Moreover, a robust machine learning methodology has the chances of improving my results from the noise of the automatically constructed datasets . The information available to the learner is thus the frame knowledge representation (a graph) plus the labels. This implies learning from structural information (as compared to learning from flat feature vectors). To this end, several alternatives are possible, including using memory-based learning, inductive logic programming, combinatorial algorithms and kernel methods [10]. The high dimensionality of the decision space over graphs made me decide for a Genetic Search over the space of all possible rules (Fig. 3), with the operators explained below.

As fitness function, I employed the F^*_α measure (weighted f-measure from Information Retrieval) of the classification task of recovering all selected labels on the training datasets. Because I wanted to obtain rules with higher recall that precision, I employed $\alpha = 2.0$ (recall is doubly important than precision). I added a MDL term to the fitness function to avoid over-fitting the data (by memorizing all training instances).

As operators, I have a sexual reproduction operator that takes the two rules and combine them in a new one, by traversing simultaneously both rule trees and randomly deciding whether to combine both rules or stick to one of the parents. I also have several mutation operators that disturb an already existing rule. As initial population, I employ rules enumerated as described in Sec. 3.1.

`TRUE()` Always select.

`IN(`**list of atomic values**`)` Select if the value is in the list.

`TRAVERSE(`**path in the graph,rule**`)` Select if the node at the of the path are selected by the rule.

`TRAVERSE-EQ(`**path in the graph**`)` Select if the value of the node at the end of the path is equal to the value of the current node.

`AND(`**rules**`)` Select if all the rules select the current node.

`OR(`**rules**`)` Select if any of the rules select the current node.

Fig. 3. Rule Language. All rules are of the form f : node $\rightarrow \{T, F\}$, that is, they take a node in the knowledge representation and return true or false.

4 Experiments

I perform two rounds of training and testing. In the first round, I use 102 biographies from `biography.com` and associated framesets from E! Online. Of these 102, 11 were separated and tagged by hand as test-set.[6] In the second round, I used 205 biographies from ImDB. From these 205, 14 were separated as test-set. Some of the rules I mined are shown in Fig. 4.

name first `TRUE()`; **name last** `TRUE()`

Always say first and last names.

education place country `IN("Scotland","England")`

As I used U.S. biographies, the country of education is only mentioned when it is abroad.

significant-other #TYPE `IN("c-husband", "c-wife")`

Mention husband and wives (but not necessarily boyfriends, girlfriends or lovers).

Fig. 4. Learned rules.

My experimental results are summarized in the following table, where I see the goodness of different approaches in the task of recovering the classification labels. I measure precision, recall and $F_{0.5}$-measure:[7]

[6] While my system uses heuristics to approximate the selected data and then learns to select that approximated selected content, the final evaluation is not heuristic: a human judge analyzes each of the kept-aside texts and marks each piece of data as appearing or not in the text.

[7] I used $F_{2.0}$ for the fitness function, but I report the quality of the system with the standard $F_{0.5}$. Results with $F_{2.0}$ will be higher, but difficult to compare.

Experiment	**development**				`imdb.com`			
	Selected	Prec.	Rec.	F*	Selected	Prec.	Rec.	F*
random	162	0.29	0.48	0.36	369	0.25	0.50	0.33
select-all	1129	0.26	1.00	0.41	1584	0.23	1.00	0.37
previous work[2]	550	0.41	0.94	0.58	891	0.36	0.88	0.51
only exact match	359	0.64	0.61	0.62	432	0.48	0.65	0.55
combined	292	0.57	0.81	0.67	432	0.49	0.68	0.57
test set	293	-	-	-	369	-	-	-

The table shows that my new system outperforms our previous work and two baselines. As the selected data can be further filtered, recall is more important than precision for my task. My combined exact match and statistically trained supervised learner obtains higher recall than the exact match trained learner, with a small penalty in precision. The `imdb.com` corpus is less homogeneous than `biography.com`, making it a harder test set, although I still observe an improvement over our previous system.

I also performed an evaluation of my Dataset Construction step: I put together all test and training corpus (102 Text-Data pairs) and obtain labelled datasets as described in Sec. 3.1. I then separated the 11 datasets corresponding to the test set and evaluated its goodness. I evaluated the exact match obtained dataset separated from the statistically obtained one. I also evaluated the two ways to combine them (one of them with three different thresholds).[8]

Exp.	Exact Match	Statistical	Full Union	Sel. 20%	Sel. 30%	Sel. 40%
Prec.	**0.75**	0.36	0.68	0.73	0.70	0.67
Rec.	0.64	0.13	**0.70**	0.69	0.61	0.53
F^*	0.69	0.19	0.69	**0.71**	0.65	0.60

The table shows the level of noise of the statistical selection process. When directly joined with the exact match data, the precision drops sharply, but there is new training material available in that dataset that cannot be captured by the exact match. The overall effect is that recall grows and there is no impact on F^*-measure. The selected union remedies that, picking the best from both datasets.

Comparing both tables I can also see that my system outperforms in recall its training data. I therefore conclude that my supervised learning is generalizing for my task beyond its initial training material, a very desirable goal.

5 Related Work

The CS literature in NLG is quite vast [11–14, 1] highlighting the importance of the problem.[9] One of the most felicitous CS algorithms proposed in the literature

[8] This is not akin to evaluate on the training corpus, because the labelling task is unsupervised.

[9] For a more thorough discussion of related work see [2, 15].

is the spread activation used in the ILEX project [1], where most salient pieces of data are first chosen (by hard-coding its salience in a field of the knowledge representation) and coherence is used to later select other data. My approach can be thought as empirically grounded mean to obtain the most salient pieces of data. This salience problem is also addressed in a summarization context [16], where human annotation are employed to provide salience scores used later for the evaluation of summaries.

The work of Reiter et al. [13] addresses also Knowledge Acquisition for CS. Different from us, although, they employ non-automatic, traditional Knowledge Engineering (and comment about how laborious is such task).

Text-Data corpora are recently gaining momentum as a resource for Statistical NLG [17–20]. They have been employed for learning elements at the content planning [17], lexical choice [18, 19] and other levels [20].

Working on the problem of learning similar substructures in graphs, the novel field of Graph Data Mining [10] is related to my supervised learning task. I think the recent invention of graph kernels [21] is of particular importance for the learning of CS logic.

6 Conclusions and Further Work

I have presented an improved method for learning CS rules, a task that is tedious to perform manually and is domain dependent. My new system improves on previously published results on two datasets.

I have mined these CS rules from a Text-Data corpus by means of Indirect Supervised Learning. This two-step learning approach is conceptually easy to grasp and it may apply to other learning from Text-Data corpora problems.

My new statistical test mines words that are related to a particular concept or concepts in the input data. In further work, I would like to relate this to ongoing research on the acquisition of linguistic data from non-linguistic sources [20] and bootstrapping learning of named-entity tags [22].

Finally, I would like to compare my Supervised Learning solution (Stochastic Search) with a kernel method, most specifically graph kernels [21].

Acknowledgements

The author wish to acknowledge the support of ARDA through AQUAINT contract MDA908-02-C-0008. He is also thankful to Kathleen McKeown, Vasileios Hatzivassiloglou, Michel Galley, Noemie Elhadad, and Juan Lipari for thoughtful comments and discussion. He is especially thankful to the three anonymous reviewers of the INLG conference.

References

1. Cox, R., O'Donnell, M., Oberlander, J.: Dynamic versus static hypermedia in museum education: an evaluation of ILEX, the intelligent labelling explorer. In: Proc. of AI-ED99. (1999)

2. Duboue, P.A., McKeown, K.R.: Statistical acquisition of content selection rules for natural language generation. In: Proc. EMNLP, Sapporo, Japan (2003)
3. Kim, S., Alani, H., Hall, W., Lewis, P., Millard, D., Shadbolt, N., Weal, M.: Artequakt: Generating tailored biographies with automatically annotated fragments from the web. In: Proc. of the Semantic Authoring, Annotation and Knowledge Markup Workshop in the 15th European Conf. on Artificial Intelligence. (2002)
4. Schiffman, B., Mani, I., Conception, K.: Producing biographical summaries: Combining linguistic knowledge with corpus statistics. In: Proc. of ACL-EACL. (2001)
5. Radev, D., McKeown, K.R.: Building a generation knowledge source using internet-accessible newswire. In: Proc. of the 5th ANLP. (1997)
6. Teich, E., Bateman, J.A.: Towards an application of text generation in an integrated publication system. In: Proc. of 7th IWNLG. (1994)
7. Duboue, P.A., McKeown, K.R.: ProGenIE: Biographical descriptions for intelligence analysis. In: Proc. 1st Symp. on Intelligence and Security Informatics, Tucson, AZ, Springer-Verlag (2003)
8. Knott, A., O'Donnell, M., Oberlander, J., Mellish, C.: Defeasible rules in content selection and text structuring. In: Proc. of EWNLG, Duisburg, Germany (1997)
9. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: Proc. of VLDB, Morgan Kaufmann (1994) 487–499
10. Washio, T., Motoda, H.: State of the art of graph-based data mining. SIGKDD Explor. Newsl. **5** (2003) 59–68
11. Sripada, S.G., Reiter, E., Hunter, J., Yu, J.: A two-stage model for content determination. In: ACL-EWNLG'2001, Toulouse, France (2001) 3–10
12. Bontcheva, K., Wilks, Y.: Dealing with dependencies between content planning and surface realisation in a pipeline generation architecture. In: Proc. IJCAI. (2001)
13. Reiter, E., Robertson, R., Osman, L.: Knowledge acquisition for natural language generation. In: Proc. of INLG-2000. (2000)
14. Lester, J., Porter, B.: Developing and empirically evaluating robust explanation generators: The knight experiments. Comp. Ling. (1997)
15. Duboue, P.A., McKeown, K.R.: Statistical acquisition of content selection rules for natural language generation. Technical report, Columbia Univ., CS Dept. (2003)
16. Nenkova, A., Passoneau, R.: Evaluating content selection in summarization: The pyramid method. In: Proc. of HLT-NAACL, Boston, MA (2004)
17. Duboue, P.A., McKeown, K.R.: Content planner construction via evolutionary algorithms and a corpus-based fitness function. In: Proc. of INLG. (2002)
18. Barzilay, R., Lee, L.: Bootstrapping lexical choice via multiple-sequence alignment. In: EMNLP-2002, Philadelphia, PA (2002)
19. Sripada, S., Reiter, E., Hunter, J., Yu, J.: Exploiting a parallel text-data corpus. In: Proceedings of Corpus Linguistics 2003. (2003)
20. Barzilay, R., Reiter, E., Siskind, J.M., eds.: Workshop on Learning Word Meaning from Non-Linguistic Data. In Barzilay, R., Reiter, E., Siskind, J.M., eds.: HLT-NAACL03, Edmonton, Canada, ACL (2003)
21. Kashima, H., Inokuchi, A.: Kernels for graph classification. In: Proc. of Int. Workshop on Active Mining. (2002) 31–35
22. Niu, C., Li, W., Ding, Jihong Srihari, R.K.: Bootstrapping for named entity tagging using concept-based seeds. In: Proc. of HLT-NAACL, Edmonton, Canada (2003)