An Analysis of The Cortex Method at TAC 2010 KBP Slot-Filling

Daniel Chada, Christian Aranha, Carolina Monte
Cortex Intelligence
Rua da Assembléia 10, suite 3711, Rio de Janeiro, RJ, Brazil, 20011-000
{daniel.chada,christian.aranha,carolina.monte}@cortex-intelligence.com

Abstract

We describe herein the Cortex Method for automated meta-data extraction and the adaptation utilized in the 2010 Text Analysis Conference (TAC) Knowledge-Base Population (KBP) slot-filling tasks, both regular and surprise-round. We delineate the software components utilized and the abstractions put forth by the method used in the evaluation. We describe and discuss the noteworthy balance shown by our results between precision and recall in automated slot-filling.

1 Introduction

Precision and recall generally pose a nearly insurmountable trade-off problem in automated unstructured text enrichment. Team Cortex's submission for the regular and surprise slot-filling tasks at the 2010 Text Analysis Conference Knowledge Base Population task has obtained a significant balance between precision and recall, thus obtaining the highest scoring submission for these tasks at the conference. Within the acceptable proprietary constraints, we hope to describe as well as possible the Cortex Method for automated text enrichment and its adaptation for the slot-filling task. We argue that the combination of linguistic rules, evidence based entity extraction, flexible knowledge representation and semantic constraints provide an exceptionally fit mix of features for such tasks.

We first describe the three main technologies behind the Cortex Method, their most salient features and the main characteristics of their interaction. In sections 3 and 4 we then provide a description of the adapted system created for the KBP slot-filling tasks. Finally in section 5 we will briefly discuss the results obtained, and argue that the balance obtained in both tasks corroborates the strength of our semantic rules in conjunction with flexible knowledge representation.

2 The Three Systems

The system used by team Cortex is an adaptation of its three main systems, using the three pillars of every Cortex text mining solution. These three independent but highly interactive systems each represent one of the initiatives of the Cortex Method: reliable recognition and classification, layered analysis and inference and, finally, flexible knowledge representation. These concepts are materialized in the three systems which implement their respective abstractions. The three systems are named: the Heuristic Evidence-and-Resource-Based Extraction (HERBE), the Summon System and the Noun Tree structure.

2.1 HERBE: Evidence-Based Entity Extraction

The Heuristic Evidence-and-Resource-Based Extraction (HERBE) is a tool for named entity identification and classification. We define by *identification* the determination of the initial and final offsets of an entity or term; and by *classification*, the label associated according to its internal ontology. Like its predecessors, HERBE utilizes a pipeline of enrichment tasks, processed in sequence, to arrive at final results.

Its distinguishing features are the use of two sources of processing, the first being a collection of

simple grammatical patterns to be detected within the text, and second the Noun Tree structure, for knowledge of relations and classification used for its decision process. These grammatical patterns are simple, and rely on local, mostly sequential knowledge to accomplish their task, as opposed to the Summon system's patterns (see below).

In HERBE each pattern has a weight associated to the precision with which it assigns an identity or class to a target entity. Local patterns (in-text patterns), in general, are stronger than global rules (based on the Noun Tree's accumulated knowledge). After the application of available patterns, the system tallies the alternatives, choosing those with the best score, considering the constraints of semantic coherence for co-reference (provided by the Noun Tree).

It is the sole scope of HERBE to provide its client systems with clear identifications of the following: *i*) identification; *ii*) collocation pinpointing; *iii*) entity chunking; *iv*) first-level classification; *v*) acronym pinpointing *vi*) first level disambiguation (sentence boundaries, paragraph boundaries, punctuation roles); *vii*) normalization.

HERBE has a certain overlap of functions with its client system, Summon: It provides an initial attempt at co-reference resolution which Summon uses as a base case for correction and expansion. Another overlapping function is that of grammatical classification. While most part-of-speech rules are contained and resolved within HERBE, the Summon system may alter or replace any instances of disagreement with HERBE. This is especially important in English, where the classification of a certain word as a verb or a noun depends more on its relations than its position.

HERBE was finalized, and replaced its predecessor Cortex EE, just in time for TAC 2010, which made the evaluation its first true field-test. While we attribute a great deal of the overall performance of the combined systems to this new evolution, posterior analysis of our results showed instances where the Summon system was unable to properly assemble sentence and text graphs (see below) and match rule-patterns due to first-level disambiguation faults on documents with unusual paragraph and line breaks.

2.2 Summon: Syntactic Relations, Anaphora Resolution and Semantic Constraints

The Summon system comprises automatic syntactic analysis, anaphora resolution and semantic constraint based rules. This system consumes the word sequence generated by HERBE (or previous versions of Cortex's EE systems), and transforms it into a word graph, in which every word is connected to others over which it exerts a syntactic relation. This system is not phrase-based, as in generative grammar models developed through Noam Chomsky's theory(1), but herein connected words possess a syntactic relation to one another.

Syntactic rules operate over word sequences in order to relate them syntactically. This allows for the use of flexible and concise rules and rule-sets, since every word class only 'knows' rules that apply to the relations it might form. As a brief example: a noun 'knows' it may relate to an article that precedes it or to another noun which succeeds it mediated by a genitive case in the sequence, but it does not 'know' it can be the subject of a verb that succeeds it. This knowledge is the scope of the verb.

The main difficulty of this approach lies in the sequence in which the relation rules should be summoned and applied. Processing cannot be based solely on the sequence in which words appear in the text, in other words, processing the rules starting with the first word, then the second, third and so on will not produce desirable results.

In order to summon the desired relations, we have developed a process of word sequence control where we may walk forward or, at times, backward in the word sequence. The system also has a stack of rules, in order to calculate dependent relations between words and word sequences.

The third step, still within the confines of the Summon system, adds further knowledge to the graph, which gains relations beyond the syntactic. Words become connected if they possess an anaphoric relation to a previous word (as in the case of pronouns).

> **"They're certainly following the predicted pattern, building Alice Dellal up more and more with stories like this (. . .) (*and further down the page*) . . . One of her aunts, Suzy, died from a heroin**

overdose while studying in Paris."

Beyond pronominal co-reference, the system infers references through parts of a name:

> **"New York City Opera has commissioned American composer Charles Wuorinen (...) (*and further down the page*) ... Wuorinen, 70, said in a statement."**

Through this co-reference, for example, the system was able to fill the age slot for Charles Wuorinen. The semantic constraints dependency resolution rule stack, working in conjunction with flexible hierarchical knowledge representation, allows for still more complex inferences:

> **"Madonna won a court battle Monday against a British tabloid that published pictures recently of her wedding eight years ago (...). The singer's adopted 3-year-old son, David Banda ..."**

The rule here may ask the Noun Tree (see 2.3 for a description or the knowledge representation system) whether any singers were previously mentioned in the text, and from that relation we find that Madonna is the mother of David Banda. The Summon system is a mature and stable system for the enrichment of English language text. Future strategic endeavors include a deep structural integration between this system and its knowledge server: the Noun Tree data structure.

2.3 Noun Tree: Flexible Representation

The Noun Tree is a flexible, hierarchical graph data structure that allows for an unlimited number of semantic relations between named entities and classes. Its main structural characteristics are that it is acyclic, directed, and allows multiple parents to any node. We still call it a tree, since "Noun Directed-Acyclic-Graph-With-Multiple-Parents" gets a bit long. Nodes in this hierarchical graph are called (very imaginatively) Noun Tree Items or NTIs. NTIs are proper nouns or named entities, or common nouns or classes. The Noun Tree serves as a lexicon and repository of nouns and relations for both the HERBE system and the Summon system.

The Noun Tree has internal logic which prevents the formation of cycles, direct-and-indirect relations between the same two nodes, and internally adapts to maintain its semantic consistency.

Two examples of such internal logic are the adaptation of relations to new insertions. Suppose NTIs c and p already exist in the Noun Tree and have a parent x in common, e.g., c IS_A x and p IS_A x. If a new relation is inserted between c and p, so that c IS_A p, the IS_A between c and x is removed. Since c is now a p and p is an x, c is, transitively, an x, without the need for a direct relation. Likewise, if an NTI x is a common child of both NTIs c and p and a new relation between c and p is created, making c a child of p, the direct relation between x and p is removed, for x is, transitively, already a p. Figure 2 depicts these properties.

Another characteristic of the Noun Tree is the capacity to dynamically associate or disassociate variant lexical items to an entity. This provides its client systems with a spectrum of possibilities for disambiguation, both of named entities and of classes.

While deeper specifications of Cortex's core systems fall into proprietary software constraints, we feel this was an adequate overall description of their underlying mechanics. In the following sections we describe the adaptations made for TAC KBP 2010, and the resulting characteristics of the collective system.

3 The application

The system developed for both the regular and surprise slot-filling tasks comprised an adaptation of the main systems of the Cortex Method to better suit the necessary demands of the slot-filling task, along with wrapper systems for input and output mapping, and satellite tasks.

Prior to the evaluation period, we used both our extensive English language news corpus (of over 3 million documents) and the subset of the TAC corpus containing news and blog documents. These were used in order to enrich the Noun Tree structure and discover the best semantic constraints to use as the base for slot-filling rules for each slot type. While the population of our structured knowledge base using the corpora was extensive (especially over Cortex's own corpus), no document in

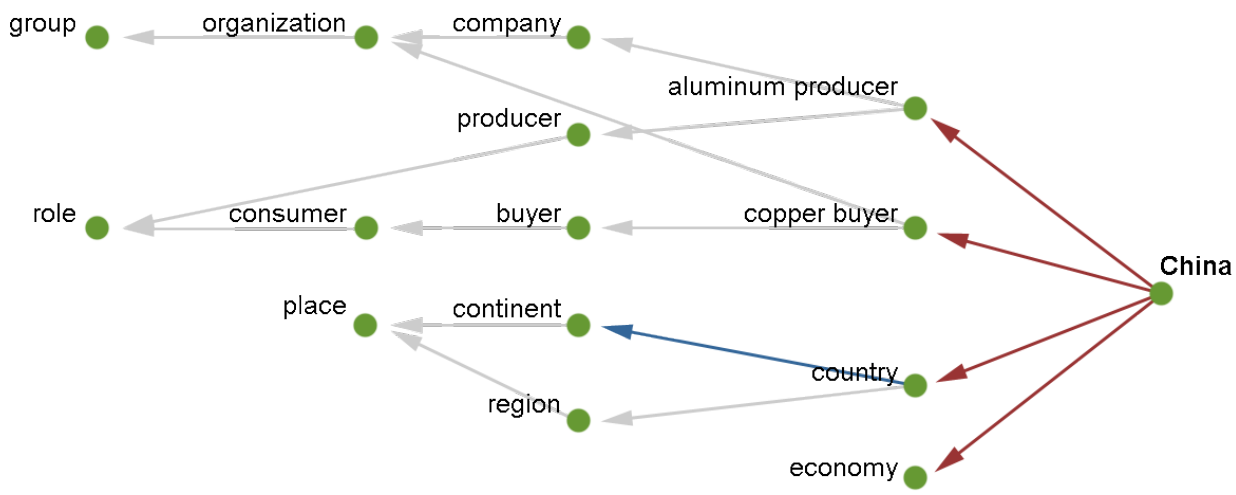

Figure 1: Visualization of Noun Tree Item 'China' and its parent relations to other NTIs. The different arrows indicate different semantic relations

either corpus was previously annotated manually for use by the system. Since the Cortex Method of textual enrichment does not use pre-annotated corpora, human effort was expended on building an appropriate set of semantic constraint rules rather than on manual annotation.

4 Slot Filling

The automated processing part of our slot-filling systems was quite similar. Both included indexing, pre-processing, identification, extraction, analysis-and-inference and slot-filling steps. In the surprise round further steps were included to provide an interface for human intervention in the results.

For both slot-filling tasks, we focused solely on our systems' domains of expertise: news and blog posts. Proposals to adapt the system to any of the other domains available in the corpus were impossible within the time frame of TAC. The structured and descriptive nature of these formats (news articles and blog posts) is, at this time, essential to our system's engines. This is important to note since the TAC corpus comprises a number of text formats, including telephone conversations and broadcast interviews, which were programmed to be ignored by our systems.

We pre-processed every news and blog post in the corpus, populating entities and variants into the Noun Tree. This, along with indexing every entity of the corpus found, permitted focusing only on the documents which actually talked about the target entities. We chose this strategy to assuage the effects of the far more time consuming slot-filling process. Additionally, pre-processing was a fundamental step in identifying misspellings, alternate names and aliases, besides populating the Noun Tree with those.

Furthermore, a great deal of effort was expended on identifying and capturing middle-eastern, Chinese, and Cyrillic entities and their variant transliterations. While we expect these efforts will aid our long term goals for the English language, they were of little use during this evaluation since the entities requested tended much more towards the Anglo-Saxon. Post-hoc analysis was worrisome, in fact, for we identified entities of Korean descent, upon which we had placed no effort at all. Specifically, we hypothesize Samsung and its slots could have fared much better with a relatively small context-specific effort. The variant transliteration system was a hybrid of a phonetic root substitution system and a modified fast Levenshtein distance calculator.

Unhandled variances in transliteration affected both recognition - where different spellings that could not be recognized were not caught; and classification, for the absence of these filters while searching the Noun Tree returned no results.

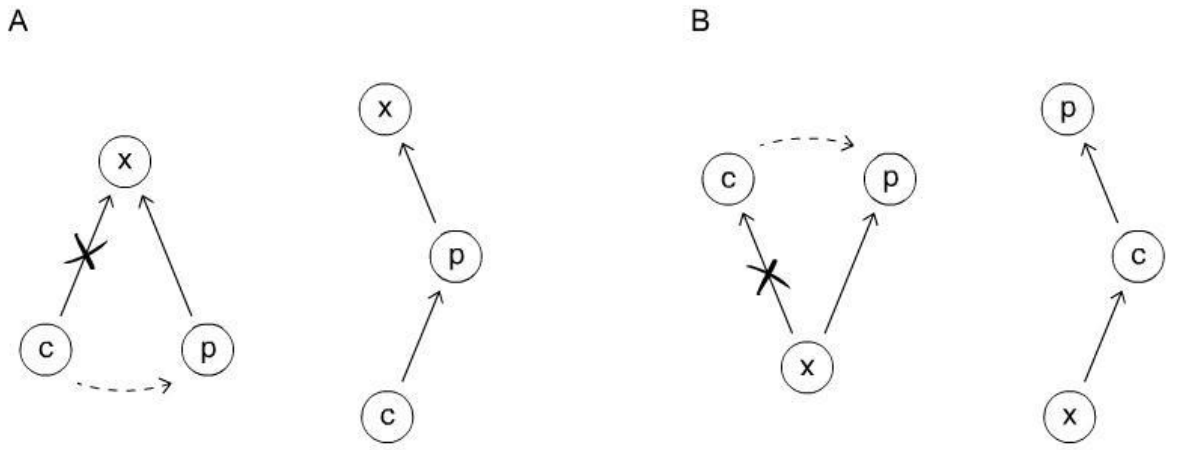

Figure 2: Two examples of automated self-updating relations in response to a new insertion.

4.1 Regular Slot-filling

The automated slot-filling comprised a new layer of semantic constraints added over the Summon system. Slots were populated according to semantic rule-patterns which had to match some sub-graph of the text. The sum of all semantic slot rules reached just over two-hundred constraints.

The adaptation of the constraints also included the necessity of guiding Summon's inferential capacities to the restrictions posed by the task rules. The clearest examples are the PER:EmployeeOf and PER:MemberOf pair (whose dichotomy proved difficult to semantically model), the PER:Title slot and the ORG:Subsidiaries and ORG:Members duo.

Once the main Summon graph was built, using semantic constraints for finding slots was a matter of interacting with the Noun Tree to find subgraphs within the graph that represents the whole text. Figure 3 exemplifies a graph representing a stateorprovince_of_birth rule:

In this context, the sentence "Chris Dodd, the Democratic Senator from Connecticut, has been rolling across Iowa" is represented by Figure 4. Within this sentence graph, we can find a pattern matching the rule in Fig. 3, which allows for the generation of a slot-relation, as in Figure 5.

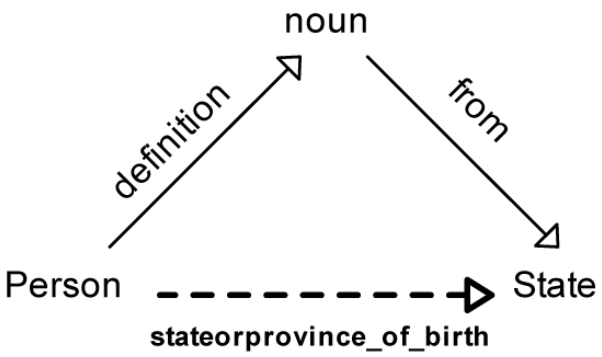

Figure 3: A graphical representation of a stateorprovince_of_birth rule.

4.2 Surprise Slot-filling

The surprise slot-filling required an adapted version of the regular slot-filling system with two main differences: the absence of slot-specific semantic constraint rules and the use of post-hoc human intervention.

We prepared for the absence of slot-specific semantic constraints by allowing for human intervention at the last step of the whole process. Prior to processing, we developed very generic grammatical and semantic restrictions based on a small number of patterns in the corpora once the slots were provided, which, as expected, resulted in a broad recall of information. This recall was processed and an interface was created to provide human-intervention

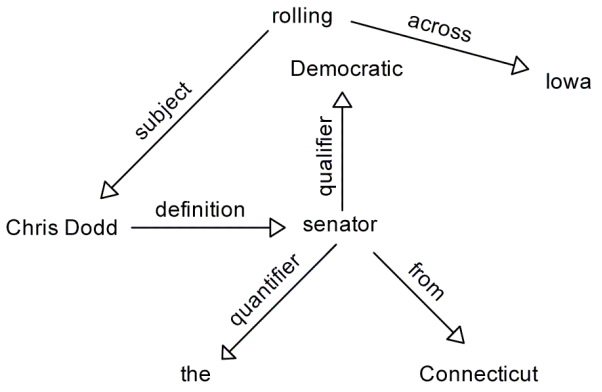

Figure 4: The resulting graph containing syntactic and semantic relations.

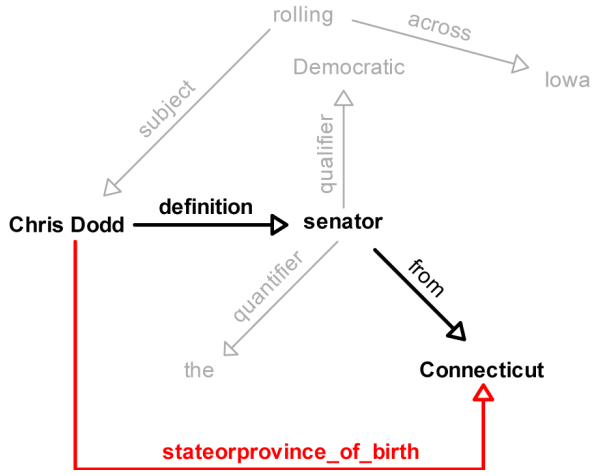

Figure 5: The generation of a slot-relation from a rule sub-graph.

in the form of a pass-fail interface. This hybrid of automated slot-filling with human judgment was our response to the time-constrained adaptation required by the surprise task.

While not fully automated, this adapted system provided similar results to the regular slot-filling. We will discuss our conclusions pertaining to the performance of the systems in their respective tasks.

5 Brief Discussion of the Results

The Cortex Method extracted the top scoring overall result in both slot-filling tasks. The noteworthy characteristic of our score lies in a high recall. For the regular task, Table 1 shows that while the difference between the Cortex Method score and the second best score was only 0.007 in precision, the second score shows a distinctly lower recall, which we suppose was an unavoidable trade-off to achieve a strong precision score.

Our results show a distinct high recall that is characteristic of a system highly attuned to relevant results in every slot of the ontology, providing strong evidence to the robustness of the combination of syntactic analysis with semantic constraints in textual enrichment.

We attribute the strong balance achieved in the regular task to the empirically tested, slot-specific semantic rules constructed through the interaction between the Summon system and the Noun Tree. This interaction allows complex structured syntactic-semantic constraints with high conceptual abstraction to be dynamically inserted, evaluated, edited and removed.

While we gave the surprise round result its due importance, we choose to interpret it as a corroboration of the strength that syntactic-semantic constraint rules and flexible representation provide to an automated system. The highly similar score provided by the *de-facto* substitution of syntactic-semantic constraints with human intervention is the strongest indicator of the strength of this collection of concepts. Table 2 describes the final scores provided for the surprise slot-filling KBP task.

6 Conclusion

The adaptation of the Cortex Method and its systems to the slot-filling tasks of TAC's KBP track proved highly successful. The systems showed a distinct precision on recall that suggests a very strong capacity for relevant profiling of entity meta-data.

We conclude that well-tuned semantic constraints are the key to our results. We look forward to continue improving the strength of the Cortex Method within this domain, and testing its potential application in other domains such as textual entailment, canonical question answering, event extraction and summarization.

References

Chomsky, N. and Lasnik, H. (1993) Principles and Parameters Theory, in Syntax: An International Handbook of Contemporary Research, Berlin: de Gruyter.

Regular Slot-Filling	**LDC**	**Top Score**	**2nd Score**	**Median**	**Cortex**
PRECISION	0.7013802	0.667996	0.6655173	0.21414538	**0.667996**
RECALL	0.54061896	0.64796907	0.18665378	0.10541586	**0.64796907**
F-MEASURE	0.6105953	0.6578301	0.29154077	0.14128321	**0.6578301**

Table 1: Results for the regular KBP Slot-filling task at TAC 2010

Surprise Slot-Filling	**LDC**	**Top Score**	**2nd Score**	**Median**	**Cortex**
PRECISION	0.8531746	0.69215685	0.52360517	0.5032258	**0.69215685**
RECALL	0.42574257	0.6990099	0.24158415	0.15445544	**0.6990099**
F-MEASURE	0.5680317	0.69556653	0.3306233	0.23636363	**0.69556653**
TIME (hrs)	N/A	99	34	11	**99**

Table 2: Results for the surprise KBP Slot-filling task at TAC 2010