Automatic text classification and property extraction

Applications in medicine

Anton Kolonin
Aigents Group
Novosibirsk, Russia
akolonin@gmail.com

***Abstract*—The paper describes different cases of combination of statistical, rule-based and pattern-based approaches for text classification, entity extraction and discernment of entity properties applied to medical domain.**

Keywords—text; mining; classification; entity; property; extraction; machine; learning; patterns; rules

I. Introduction

There are numerous approaches to text mining involving techniques for automatic text classification, entity extraction and discernment of the entity properties [1], including those applied to medical domain. They vary based on extents that they rely on different techniques such as statistical machine learning or pattern matching or natural language parsing. Respectively, they can be described with different accuracy and run-time performance with typical trade-off between the former and the latter, so that high recall and precision of results typically are connected with low processing throughput and vice versa. In this paper we attempt to describe somewhat integrated approach involving statistical techniques along with use of patterns and rules, possible extensible to deal with complete natural language processing. The approach has been evolved over ten years in course of few different projects and products described further.

All discussed text mining problems (classification and extraction of entities with discernment of their properties) can be treated as specific cases of association of a concept (having certain level of abstraction) with a piece of text of certain size. That is, classification is about associating some top-level concept (like "microbiology" or "gynecology") with entire paper or article. In turn, entity extraction is about finding out more specific concept, like some person or clinical case spanned over one or few text sentences in entire text. Finally, discernment of entity property refers to identification of very specific concept (like name or gender of a person or symptom of a clinical case) within spot of text identifying particular entity in certain phrase.

II. Approach

Typically, the techniques used for text mining could be classified as: 1) statistical machine learning methods, relying on various feature formation methods and statistical models for operating with feature vectors; 2) "shallow parsing" or rule-based techniques based on exhaustive set of templates and rules manually engineered for specific domain; 3) complete text comprehension based on linguistic model for a specific language and given ontology (again, engineered for specific domain) "grounded" on that language.

In our view, the latter two can be considered as different extremes of the same, "generic pattern-based" approach. To deal with patterns, in the further discussion, we will be using simple notation [2] for different sorts of sets, N-grams and syn-sets, where parentheses identify conjunctive set of features (like "(a b c)" means "a AND b AND c"), curly braces identify disjunctive set of features (syn-set like "{a b c}" means "a OR b OR c") and square brackets identify ordered conjunctive sets or "N-grams" (so "[a b c]" describes "a → b → c") .

That is, there are simple patterns such as series of words or tokens associated together in two ways. First, there are sequences or N-grams to be matched in order (such as "[cancer treatment]"). Next, there are "syn-sets", associating several alternatives – be it semantic "syn-set" such as "{treatment, cure}" or different linguistic forms of the same dictionary word such as "{ill, illness} or "{sick, sickness}" or both mixed together. Also, the patterns can include wildcards, regular expressions or placeholders (while the placeholders can have specific domain restrictions, based on custom dictionaries or specific rules). Further, complexity of the patterns could be increased making it possible for N-grams and syn-sets to include other patterns, building hierarchical representations for each of specific categories to be matched, entity property value to be discerned or an entity to be extracted. Finally, using hierarchical system of patterns, potentially, the entire natural language can be encoded with one single pattern, so that applying this pattern for text classification would enable automatic identification of entire text to be written in one or another human language.

From our perspective, having depth of patterns generally varying, the two issues arise. The questions are: "who is creating the patterns and rules?" (from one perspective) or "who is building linguistic model and ontology?" (from another perspective). In both cases, this could be done either manually (and it is known to take many human-years for a language or a domain) or learned statistically with help of machine learning (regression analysis for hierarchical patterns and rules or "deep learning" [1]). As we believe, relying on experience discussed further, the two approaches could

complement one another in practical applications. It is also assumed that simpler patterns and rules lead to higher throughput while data processing and lower cost of it while more complex ones imply getting things done in slower and more expensive way. That is, a practical solution should be engineered having this in mind, given specific problem domain, required processing speed and available computational resources.

All that said, we consider "shallow parsing" techniques using rules and patterns and natural language parsing with ontology-based comprehension to be technically the same, from the implementation perspective, with the only difference in level of pattern complexity, amount of rules and source of the two.

We also consider statistical methods to be part of the generic approach and practical framework because of the two very practical reasons. Primarily, it is practically not feasible to obtain input textual data completely error-free and matching the training corpus (used to engineer linguistic rules and/or ontology) perfectly. And then, there is a practical need to apply "fuzzy rules" [3] expressed in statistical measures – whether given category could be associated with an entire text or a property value or an entity could be extracted from specific phrase.

Moreover, while there are training corpora and linguistic models are present for most of European and few Eastern languages, many of the Earth languages (such as spoken in Sri Lanka for instance) do not have event plain locale support in Android software platform, leave alone exhaustive formal linguistic model and computable dictionary. That makes barely possible to localize a text mining software solution for particular locales, unless the solution supports statistical approach employed to evolve linguistic rules and patterns for a domain dynamically, in the course of live system interactions with native speakers (plain users or engineers).

III. Statistical Learning In "Webcat" System

The first implementation of the described approach has been done with "Webcat" text classification system as part of larger "Webstructor" project. The demonstration version of the software is available online at http://www.webstructor.net/mine/. This involves automatic text classifications based on statistical learning relying on operations with feature vectors [4], with computational model presented on the Fig. 1 and Fig 2.

In the described implementation, the list of features is restricted to simple one-word tokens. Operational graph provides two use cases, supported with two user interfaces for regular user an expert-analyst, with respective user interfaces developed.

The regular user interface provides ability to enter "documents" (as either portions of plain texts or links to local files or documents on the Web), assign features and categories to the documents, render the documents with identified features highlighted respectively to the importance of a feature to the given document and (importantly!) review the categories and the features that the system assigned to documents automatically. In the course of review, user can either mark a feature or a category as "irrelevant" (giving "negative feedback") or confirm their relevance to the document (providing "positive feedback").

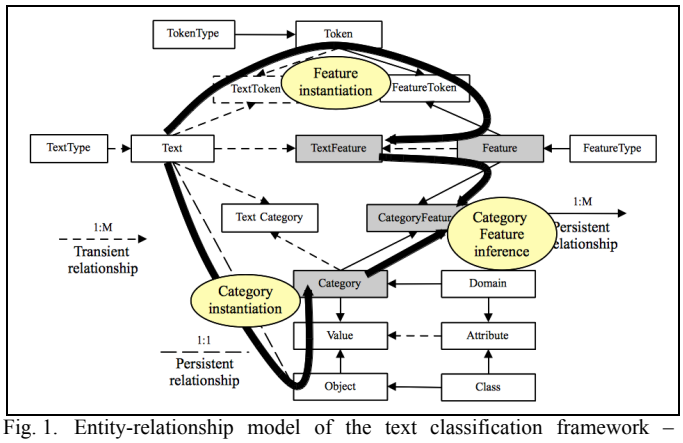

Fig. 1. Entity-relationship model of the text classification framework – learning phase. There are three sub-processes contributing to the learning process: 1) Category instantiation which takes attribute values identified for texts in training corpus (either encoded in the text as tags or taken from respective database table fields) and creates categories for them, given the category domain indicated by the attribute; 2) Feature instantiation which takes text in training corpus and decomposes it into tokens and features accordingly to the parser, tokenizer and feature builder depending on the implementation; 3) Category Feature inference which employs machine learning statistics [4] to infer relevance of features encountered in the texts to the categories associated with those texts.

The expert-analyst workplace interface provides category-centric view for user, giving them ability to browse categories along with features and documents associated with them, as well as render the documents with feature highlighting from perspective of the selected category.

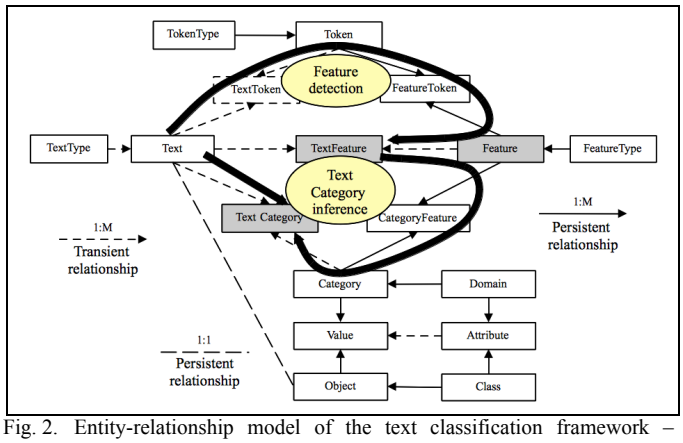

Fig. 2. Entity-relationship model of the text classification framework – recognition phase. There are two sub-processes contributing to the recognition process: 1) Feature detection which takes the text in novel data and decomposes it into tokens and features accordingly to the parser, tokenizer and feature builder depending on the implementation, - this process is similar to Feature instantiation in the course of learning, but the key difference is that only the features instantiated earlier in the course of learning can be detected, no new features are instantiated; 2) Text Category inference - employs statistics to infer the relevance of texts to the categories associated with those texts through the features detected in the texts and learned for those categories.

Similarly to the regular user, the expert-analyst can review mutual assignments of documents and features, tuning overall system classification capabilities by means of giving positive or negative feedback to it on either feature or document or both – from perspective of given category.

The feedback supplied by both kinds of users is internally used by the system to update statistical measures and update classification results at run-time.

To enable greater precision and segmentation of the problem domain, documents can be allocated in multiple separate "document sources" while categories are allocated to different orthogonal "domains", so the same document can be tagged with categories from multiple domains while same category can be used to classify documents in different sources.

The Webcat system has been tested against two problems relevant to medical domain. The first case is represented by existent demo version and provides normalization of named entities (research facilities) associated with publications in PubMed resource, so that different variations of the same research facility reference in text could be matched one against another. The second case was used for classification of topics of questions on Internet forum, with training based on raw questions submitted by patients to online healthcare portal [5], made it possible to associate patient questions with specific diagnoses and doctor specializations in order to route online questions to specific doctor to handle. In both cases, the overall recall/precision have been found in range 65-85%, depending on quality of initial tagging in the training corpus and extent the data in testing corpus is matching training corpus.

IV. Adding Rules And Patterns

The former solution has been extended for the purpose of discernment of entity properties while processing textual inventory catalogs, invoices and price list records – for the purpose of identification of categories and attribute values of products and materials used in healthcare. Specific of the problem was high level of abbreviation and error rate in the original records themselves, so using any conventional linguistic or pattern-based approaches did not work. In its turn, use of statistical method did not provide recall and quality sufficient to extract couple tens attribute values from short text records. In addition, high throughput requirements were posed for production system, so it turned out that full-scale operations with feature vectors were out of the question.

In the course of solution [4] development, several enhancements were made to the "Webcat" system, introducing "sparse N-gram" features, feature prioritization, "boolean ranking", contextual scoping, compression of feature vector space and fuzzy word matching.

"Sparse N-gram" feature has been introduced to support matching regular multi-word patterns with possibility of interjecting aside tokens in the text being matched. This helped to increase recall, identifying cases when say "[cannula adapter]" pattern identifying item type had to be matched against "cannula 15mm adapter" text with interjection of "15mm" token referring to another attribute identifying item size.

Feature prioritization implied giving more decisive power to more complex features. Namely, longer N-grams were given more priority, so given an attribute value is matched with help of sequence of N elements, simpler N-grams with less elements (down to single token as 1-gram of 1 element) were not considered at all. This helped to increase precision in case like when "cannula adapter" identifies the item better than "cannula" and "adapter" alone, as they may be used in multiple contexts.

"Boolean ranking" has been found to improve precision by means of implicit consideration of conjunctive operation for the features identifying category. That is, the first-level ranking was performed relying on total number of features identifying category and then the ties were broken with second-level ranking based on statistical evidence. This has lead to further development of idea to use complex hierarchical patterns (disjunctive, conjunctive, and ordered conjunctive) discussed in the following section.

Contextual scoping technique turned helpful to increase both precision and throughput by means of using statistically inferred and manually controlled ontology, with specific "leading" ("primary") attributes identifying item type with set of other "dependent" ("secondary") attributes, so that only "dependent" attributes were processed based on "leading" ones identified in the first place. More sophisticated variant of contextual scoping involved restriction of attribute value domains based on "leading" attributes. While the latter technique did help to increase precision, the impact of it on throughput turned to be negative enough so it has not been used in production.

Compression of feature vector space involved removing low-frequency features associated with target categories and attribute values. It turned to be necessary to provide reasonable throughput, having the recall sacrificed. Indeed, it has turned out that limit on size of feature vectors used for matching can effect in dramatic (tens of times) positive impact on run-time performance while having reasonable negative impact on recall (few percent). However, this procedure introduced problems for incremental learning, because low-frequency features may be lost during compression and so once they are "compressed out", they are losing chance to get frequency increased. The latter problem has been overcome storing uncompressed feature vectors for incremental learning purposes but creating compacted feature extracts each time when high-speed input recognition were required.

Finally, to deal with noise in word tokens due to high typo rate, we tried fuzzy token feature matching decomposing each token into conjunctive set of letter bi-grams and matching such sets instead of matching strings. This technique has provided positive impact on recall however has affected throughput negatively and dramatically. In the end, the more cheap and efficient enough solution turned to be just involving variations of most frequent typos of a word into respective disjunctive sets (speculative "syn-set"), which also turned quite helpful to deal with multiple forms of abbreviations.

Described implementation have been tested against raw inventory data from hospitals – about 3 million lines of text,

each line representing particular entity with 10-25 properties (pre-processed manually by human operators both for training and testing). Obtained results provide joint recall and precision metric in range 75-95%, with ~95% corresponding to testing sets of the same origin as training sets and ~75% corresponding to testing sets from independent sources. Selective investigation of mismatches has shown that ¼ of mismatches were due to human-factor errors (typos and invalid attribute value assignments) in testing corpus, ¼ of mismatches – due to similar human errors in training corpus (so that learned feature vectors and rules have been invalidated in advance), other ¼ - due to novel data in testing corpus not supported by training corpus and final ¼ – due to inefficiency of the approach and implementation on itself (limited size of N-grams, lack of proper natural language parsing, missed contextual associations between values of different attribute, etc.) [4].

V. Hierarchical Patterns In Aigents Platform

Our current work is dedicated to creation of personal light-weight "Aigents" software system [2] for individual and corporate use, capable of selective navigation on the Internet and extraction of target information from web sites – now available for evaluation at https://aigents.com. Typical use-case of the system is having specific area or topic of interest (for example, particular set of medical products) associated with known set of Internet resources (for instance, sites of potential vendors, clients or competitors), being able to monitor appearance of specified information objects on the sites, capturing their properties and returning results to requesting user as soon as possible. In the scope of this task, there are two kinds of text mining problems to address. First, there is a need to spot specific pages of the Web sites (for instance "Products" and "Services" pages of entire site), following the Web navigation links leading to those pages and skipping irrelevant links and pages – contextual text classification is employed for this purpose [6]. Second, there is a need to determine areas identifying descriptions of target entities on spotted pages and extracting their properties – which is done my means of using patterns to locate entities and extract their properties.

For the described purpose, we developed system of hierarchical pattern matching, supporting patterns expressed in the following BNF notation [2].

<pattern> := <token> | <regexp> | <variable> | <set>

<set> := <conjunctive-set> | <N-gram> | <syn-set>

<conjunctive-set> := (<pattern> *)

<N-gram> := [<pattern> *]

<syn-set> := { <pattern> * }

That is, a pattern is comprised with sequence of elements where each element can be either textual <token> (i.e. word or literal sequence), standard regular expression <regexp> (for matching), <variable> placeholder (to be filled in) or <set>. The <set> can be either <conjunctive-set> or ordered conjunctive <N-gram> or disjunctive <syn-set> described earlier (framed with parentheses or brackets or braces respectively).

Moreover, the patterns are accompanied with extensible ontology [2] which can be used to describe taxonomy of objects of interest as well as properties associated with particular classes of objects, altogether with domains of the property values. Each variable in a pattern could be implicitly or explicitly associated with property of some object (class) in the taxonomy, and could be also given some domain restriction – either hardcoded (such as "number", "date", "time", "currency") or expressed by means of regular expression. Hence, patterns without variables could be used to tag or classify either entire texts or particular segments of the texts while patterns with variables could be used to discern particular properties of the identified entities.

To simplify management of patterns, ontology and domain restriction, dedicated language AL [2] has been employed.

For the medical domain, system has been tested for problem of marketing research [6] and supply chain management, where the task has been defined to identify specific products of interest on particular web sites. For example, for product called "catheter", different patterns have been developed for different sites. In the following examples, ontology structure, patterns and domain restrictions are presented for particular medical vendor sites – using AL language.

http://www.bardmedical.com/products/

Catheters has type, brand, ways, size, quantity, order_number, part_number, fr, length, diameter, tip. Catheters patterns '$type catheters $brand , $ways $part_number $size $quantity'. Type is word. Ways is '/^[0-9]{1}\-way$/'. Quantity is '/^[0-9]{2}\/case$/'.

https://www.cookmedical.com/products/

Catheters has type, brand, ways, size, quantity, order_number, part_number, fr, length, diameter, tip. Catheters patterns '$order_number $part_number $fr $length cm $diameter inch', '$order_number $part_number $fr $diameter $length $tip'. Order_number is '/^g[0-9]{5}$/'. Part_number is '/^[0-9a-z\.\-]{6,25}$/'. Fr is number. Length is number. Diameter is number. Tip is word.

http://www.bbraunusa.com/products.html

Introcan catheters has brand, ga, diameter, details, product_code. Introcan catheters patterns '$brand catheter $ga ga. x $diameter in., $details $product_code'. Ga is number. Product_code is '/^[0-9]{7}-[0-9]{2}$/'.

In the examples above, for each of AL description block for one of three sites, first phrase defines properties specific to the target item class using ontological "has" relationship. Second phrase identifies pattern – in our example all patterns happened to be straight N-grams consisting of either tokens or variables (tokens preceded with dollar $ sign). Finally, few last phrases in each block specify domain restrictions using ontological "is" relationship. The universal pattern dealing with text content from all three sites can be represented as follows, with hierarchical pattern of three levels.

Catheters patterns '{[$type catheters $brand , $ways $part_number $size $quantity] [$order_number $part_number $fr $length cm $diameter inch', '$order_number $part_number $fr $diameter $length $tip] [$brand catheter $ga ga. x $diameter in., $details $product_code]}'.

The approach has been successfully tested for extracting target information from the web sites in real time and storing it in semantic graph database, with 100% accuracy given required amount of human resources spent on pattern fine-tuning for particular product type represented on given web site.

VI. Conclusion And Future Work

Different text mining approaches and solutions described above appear reasonable and efficient to solve various problems in medical industry and beyond the domain. Statistical method appears valuable to build feature vector models for wide range of feature complexity, including hierarchical pattern features. However, the two major problems have been identified for future work.

The first obvious one is amount of manual effort needed to create and test respective templates for new sites and products or update them as the site structure changes. While this theoretically could be done with "deep learning" [1] technique, building hierarchical representations based on training sets supplied by users, there is still need to establish workflow involving human operator verifying and fine-tune patterns determined automatically and the results that they produce. So need to develop efficient "deep learning" techniques along with usable interfaces for training the system by means of verifying its results within continuous integrated human-machine process.

The second problem turned to be text segment boundary detection to identify spot of an object appearing in the text in generic case. The natural sentence identified by punctuation boundary might be working in some cases but failing when an object is described with help of several sentences. At the same time, natural sentence boundary on itself may be not identified with conventional punctuation if dealing with Web pages where HTML markup structure is used. To address this, we are suggesting to represent source texts in hierarchically structured way (e.g. inherited from HTML or PDF markup), and then identify the closure of the block where the beginning of a pattern matched as an implicit end of the matching text segment.

References

[1] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa, "Natural language processing (almost) from scratch", Journal of Machine Learning Research 12 (2011) 2461-2505.

[2] A. Kolonin, "Intelligent agent for Web watching: Language and belief system", Problems of Informatics, ISSN 2073-0667, Issue 2, 2015, pp.59-69.

[3] G. Akrivas, G. B. Stamou, S. Kollias. "Semantic association of multimedia document descriptions through fuzzy relational algebra and fuzzy reasoning", IEEE Transactions on Systems, Man, and Cybernetics, Volume 34, Issue 2, March 2004.

[4] A. Kolonin, "High-performance automatic categorization and attribution of inventory catalogs", Proceedings of All-Russia conference Knowledge Ontology Theories (KONT-2013), Novosibirsk, Russia, 2013.

[5] A. Kolonin, A. Volnukhin, "Analytical model of adaptive medical diagnostics", Proceedings of All-Russia conference Knowledge Ontology Theories (KONT-2013), Novosibirsk, Russia, 2013.

[6] A. Kolonin, "Machine learning of software agents working with natural language texts", Control Engineering, Russia, Issue 3(57), June 2015, pp.82-85.