

Information Extraction and Classification from Free Text Using a Neural Approach

Ignazio Gallo and Elisabetta Binaghi

Department of Computer Science and Communication
Università degli Studi dell'Insubria
via Mazzini 5, Varese, Italy
ignazio.gallo@uninsubria.it
<http://www.uninsubria.eu/>

Abstract. Many approaches to Information Extraction (IE) have been proposed in literature capable of finding and extract specific facts in relatively unstructured documents. Their application in a large information space makes data ready for post-processing which is crucial to many context such as Web mining and searching tools. This paper proposes a new IE strategy, based on symbolic and neural techniques, and tests it experimentally within the price comparison service domain. In particular the strategy seeks to locate a set of atomic elements in free text which is preliminarily extracted from web documents and subsequently classify them assigning a class label representing a specific product.

Keywords: Information Extraction, Neural Network, Text Classification

1 Introduction

With the Internet becoming increasingly popular, more and more information is available in a relatively free text format. This situation creates the premise for efficient on line services and Web mining application in several domains. The on line availability of ever larger amounts of commercial information, for example, creates the premise for profitable *price comparison services* allowing individual to see lists of prices for specific products.

However critical aspects such as information overload, heterogeneity and ambiguity due to vocabulary differences limit the diffusion and usefulness of these advanced tools requiring expensive maintenance and frustrating users instead of empowering them.

To address these problems, efficient *Information Extraction* (IE) techniques must be provided capable of finding and extract specific facts in relatively unstructured documents. Their application in a large information space makes data ready for post-processing which is crucial to many context such as Web mining and searching tools.

Information extraction programs analyze a small subset of any given text, e.g., those parts that contain certain trigger words, and then attempt to fill out a fairly simple form that represents the objects or events of interest. An IE task is defined by its input and its extraction target. The input can be unstructured documents like free text that are written in natural language (e.g., Fig. 1) or the semi-structured documents that abound on the Web such as tables or itemized and enumerated lists (e.g., Fig. 2). The extraction target

of an IE task can be a relation of k-tuple (where k is the number of attributes in a record) or it can be a complex object with hierarchically organized data.

Many approaches to IE have been proposed in literature and classified from different points of view such as the degree of automation [1], type of input document and structure/constraint of the extraction pattern [2].

This paper proposes a new IE strategy and tests it experimentally within the price comparison service domain. Most price comparison services do not sell products themselves, but show prices of the retailers from whom users can buy. Since the stores are heterogeneous and each one describes products in different ways (see example in Fig. 3), a generic procedure must be devised to extract the content of a particular information source. In particular our IE strategy seeks to locate a set of atomic elements in free text preliminarily extracted from web documents and subsequently classify them accordingly. In this context, our principal interest is to extract, from a textual description, information that identify a commercial product with an unambiguous label in order to be able to compare prices. In our experiments product price was associated to the description, therefore its extraction is not necessary.

The Motorola RAZR V3i is fully loaded* - delivering the ultimate combination of design and technology. Beneath this sculpted metal exterior is a lean mean, globe-hopping machine. Modelled after the Motorola RAZR V3, the RAZR V3i has an updated and streamlined design, offering consumers a large internal color screen, . . .

Fig. 1. An unstructured document written in natural language that describes the product 'Motorola RAZR V3i'.


Following the terminology used by Chang et. al[3] the salient aspects are

- an hybrid solution for building thesaurus based on manual and supervised neural technics
- a tree structured matcher for identifying meaningful sequences of atomic elements (tokens);
- a set of logical rules which interpret and evaluate distance measures in order to assign the correct class to documents.

2 System Overview


The IE system developed is composed of two main parts, *matcher* and *classifier*, and acts on free text documents obtained from original web documents. It specifically addresses the following problems typical of the price comparison service information space: an attribute may have zero (missing) or multiple instantiations in a document; various permutations of attributes or typographical errors may occur in the input documents (see an example in Fig. 3).

HOT DEALS



Panasonic DMC-TZ3EB-S Silver
 In stock now
 quicklinx: 4HGPTB
 mfr#: DMC-TZ3EB-S
10x Optical Zoom and 28mm Wide Angle Leica Lens
£239.99 inc vat




1 **ADD**






Olympus FE-190 Digital Camera refurb
 In stock now
 quicklinx: 4HGPTB
 mfr#: DMC-TZ3EB-S
6 M
£7

1 **ADD**





90% of our customers trusted us by choosing a pack option!


+

+






Olympus FE-230 + Sandisk xD Memory Card 2 GB + Deal Display Nylon Case - black


+

+






Olympus FE-230 + Olympus 1 GB Panorama xD memory card + Deal Display Nylon Case - black


+

+

+


Olympus FE-230 + Olympus E0413743 Leather case + LI-42B battery kit + Olympus 1 GB Panorama xD memory card


+

+

+


Olympus FE-230 + Olympus 1 GB Panorama xD memory card + EFORCE Compatible battery Li-42B + Pixmania L cm PIX leather case


+

+

+


Olympus FE-230 + Olympus 1 GB Panorama xD memory card + EFORCE Compatible battery Li-42B + Deal Display Nylon Case - black

Fig. 2. Semi-structured documents written in natural language that describes a set of products.

Prior to both matching and classification phases the *tokenizer* divides the text into simple tokens having the following nature:

- **word**: a word is defined as any set of contiguous upper or lowercase letters;
- **number**: a number is defined as any combination of consecutive digits.

2.1 Matcher

In our context the matcher has to operate on specific annotations that can be matched to brands (B) and models (M) of products enlisted in the price comparison service. The matcher is then constructed starting from a set $KB = \{(b, m) | b \in B, m \in M_b\}$ that contains the couples formed by a brand b and a model m . b belongs to the set of brands B of a given category of products, while m belongs to the set of models M_b that have b as brand.

A thesaurus that collects all the synonyms commonly used to describe a particular category of products is used to extend the KB . In particular if there are one or more synonyms in the thesaurus for a couple (b, m) then we add a new couple (\bar{b}, \bar{m}) to the KB for each synonym.

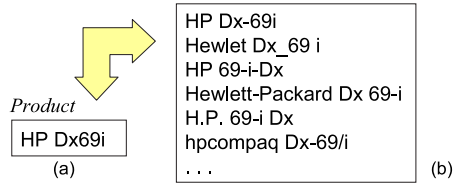


Fig. 3. List of potential descriptions in (b) that may be used to describe product (a).

The tokenizer associates with every couple in the KB a sequence of tokens $T_{b,m} = (t_1^b, t_2^b, \dots, t_1^m, t_2^m, \dots)$ where t_i^b and t_j^m are all the tokens obtained from the brand b and the model m respectively. Every sequence $T_{b,m}$ is used to construct a path within the matcher tree structure: it starts from the first node associated with token t_1^b and arrives at a leaf node associated with the label derived from (b, m) (see example in Fig. 4).

Based on these solutions, the IE task is accomplished splitting an input document into a tokens sequence (d_1, d_2, \dots) . Starting from the root node the search for the subsequent node (better match in the subsequent layer) is performed using the concept of *edit distance* (ed) between input tokens and every token of a tree's layer and the *position distance* (pd) between two matched input tokens. The *edit distance* between two tokens measures the minimum number of unit editing operations of *insertion*, *deletion*, *replacement of a symbol*, and *transposition of adjacent symbols* [4] necessary to convert one token into another. The *position distance* measures the number of tokens d_j found between two matched d_j d_k tokens that would have to be consecutive

The system first searches a sequence of perfect match ($\sum ed = 0$) that leads to recognition of all the tokens t_i^b associated with one brand. In this way we obtain a list of possible brands associated with the input document. Starting from the last node (with token t_i^b) of a matched brand, the algorithm begins the search for the most probable model. The system searches all the paths that lead to a leaf node with the sum of ed equal to zero. If this is not possible the system returns the path with minimum ed .

In case of a token t_i with multiple match (d_j, d_k, \dots) , with identical minimum ed , the input token with minimum pd (compared to the parent token in the tree) will be selected (see example in Fig. 5).

2.2 Classifier

A rule based classifier was designed with the aim of assigning a class label representing a specific product to each document using the information extracted from the matching phase.

The classifier receives in input the output of the matcher i.e. the set of matched sequence $T_{b,m}$ of tokens weighted as a function of the ed . These input values are used to construct a sub-tree of the Matcher starting from which the classifier computes the class of the given document.

The set of predefined classes is constituted by all the products (b, m) inserted in the KB . The classifier selects a class from a subset obtained by the input set of matched sequence $T_{b,m}$ (there is one class for each matched sequence).

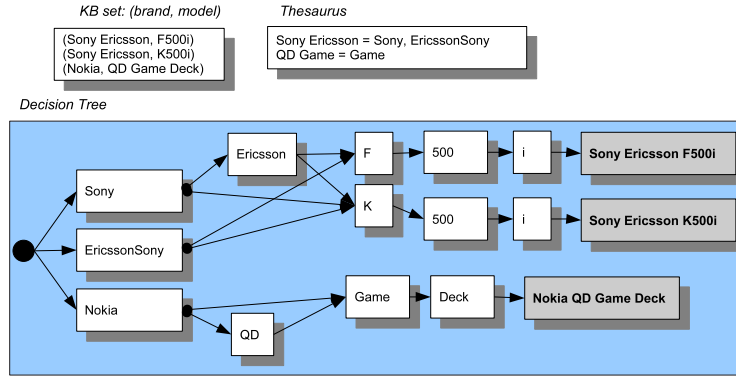


Fig. 4. An example of *KB* set and thesaurus used to build the matcher.

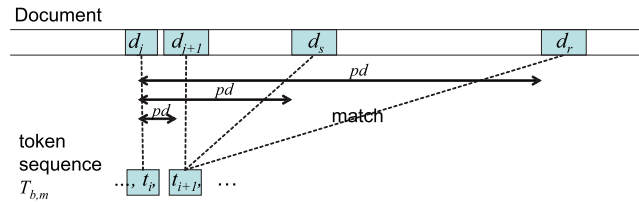


Fig. 5. Role of edit distance and position distance during the matching process. In case of tokens having equal *ed* measure ($ed(t_{i+1}, d_{j+1}) = ed(t_{i+1}, d_s) = ed(t_{i+1}, d_r)$) the token d_{j+1} with minimum *pd* is selected.

Position and edit distances are evaluated to decide class assignment: the classifier start from the leaves of the sub-tree and at each step compares each node t_{ij} with its parent t_{i-1} using the following rules:

- select the node t_{ij} having minimum $pd(t_{ij}, t_{i-1})$ for each j or with minimum average *pd* computed through all the seen tree nodes;
- in case of a node t_{ij} with multiple match associated with its parent t_{i-1} , with identical minimum *ed*, the input token t_{i-1} with minimum *pd* will be selected (see example in Fig. 6);
- between two nodes t_{ij} with identical weight ($ed + pd$) in the same layer i , select that with a greater path starting from the leaf node;
- if all the previous rules find no differences between two nodes t_{ij} and its parent t_{i-1} then selects that with minimum *pd* computed by the matcher.

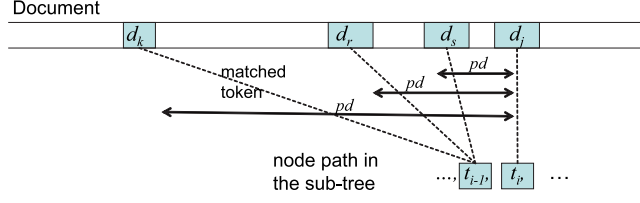


Fig. 6. Role of edit distance and position distance during the classification process. In case of tokens having equal ed measure ($ed(t_{i-1}, d_k) = ed(t_{i-1}, d_r) = ed(t_{i-1}, d_s)$) the token d_s with minimum pd is selected.

3 Automatic Thesaurus Building

The accuracy strongly depends on the completeness of the thesaurus. Unfortunately, thesaurus maintenance is an expensive process. The present work proposed a neural adaptive tool able to support thesaurus updating.

The main idea of our solution is to automatically induce from examples general rules able to identify the presence of synonyms within sequences of tokens produced by the matcher. These rules, difficult to hand-craft and define explicitly, are obtained adaptively using neural learning. In particular, a Multilayer Perceptron (MLP) [5] is trained to receive in input the following types of features from the sequence of matched tokens:

- edit distance between the tokens
($t_1^b, t_2^b, \dots, t_1^m, t_2^m, \dots$) of the KB and tokens found in the document;
- number of characters between two consecutive tokens found in the document;
- typology of found tokens (word or number);
- bits that identify the presence of each token.

As illustrated in Fig. 7, each pattern is divided into four groups of P features, where P represents the maximum number of tokens obtainable from a product (b, m) .

The output pattern identifies a synonym \bar{S} of S , where S is a particular sequence of tokens extracted from $T_{b,m}$, and \bar{S} is a different sequence of tokens extracted from $T_{b,m}$ or from the sequence of matched token of the document. Each output pattern has a dimension equal to $3P$. The output of the trained neural network is used to add a new item $S = \bar{S}$ to the thesaurus: when an output neuron has a value greater than a threshold, the corresponding token will be considered part of a synonym.

4 Experiments

The aim of these experiments was to measure the classification effectiveness in terms of precision and recall, and to measure the contribution of neural thesaurus updating within the overall strategy.

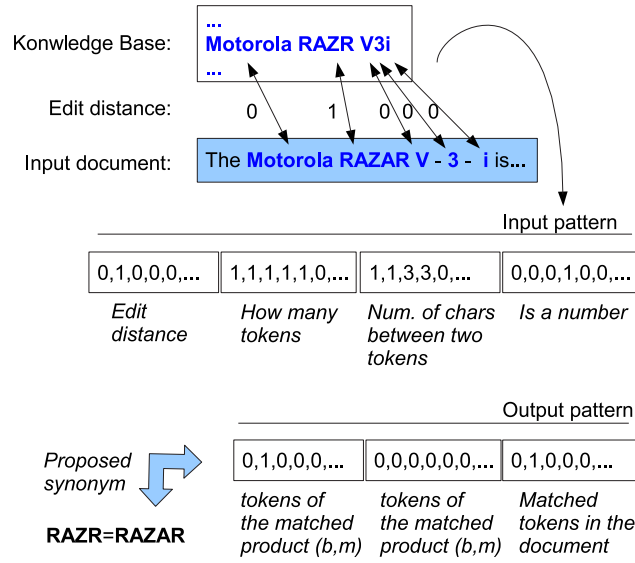


Fig. 7. An example of input pattern/output construction for the MLP model.

4.1 Text Collection

Document collection is taken from the price comparison service Shoppadoo (<http://uk.shoppadoo.com> and <http://www.shoppadoo.it>), a price comparison service that allows users to compare prices and models of commercial products.

Two experiments were conceived and conducted in the field of price comparison services. Two product categories were identified, *cell-phones* and *digital-cameras*. Three specific brands were considered in our set of couples (b, m) KB for each category (Nokia, Motorola, Sony for cell phones and Canon, Nikon Samsung for digital-camera). The total number of documents collected for the cell-phone category was 1315 of which 866 associated with one of the three identified brands. The number of documents belonging to the digital-camera category were 2712 of which 1054 associated with one of the three identified brands. Remaining documents belonging to brands different from those considered in the experiment, must be classified *not relevant*.

4.2 Evaluation Metrics

Performance is measured by recall, precision and F-measure. Let us assume a collection of N documents. Suppose that in this collection there are $n < N$ documents relevant to the specific information we want to extract (brand and model of a product). The IE system recognizes m documents, a of which are actually relevant. Then the *recall*, R , of the IE system on that information is given by

$$R = a/n \quad (1)$$

and the *precision*, P , is given by

$$P = a/m \quad (2)$$

One way of looking at recall and precision is in terms of a 2×2 contingency table (see Table 1).

	Relevant	Not-relevant	Total
Matched	a	b	$a + b = m$
Not-matched	c	d	$c + d = N - m$
Total	$a + c = n$	$b + d$	$a + b + c + d = N$

Overall accuracy (OA): $(a + b)/N$

Table 1. A contingency table analysis of precision and recall.

Another measure used to evaluate information extraction that combines recall and precision into a single measure is the F-measure F_α defined as follows:

$$F_\alpha = \frac{1}{\alpha \frac{1}{P} + (1 - \alpha) \frac{1}{R}} \quad (3)$$

where α is a weight for calibrating the relative importance of recall versus precision [6].

4.3 Results

The results of the two experiments are summarized in Tables 2 and 3. We started with an empty thesaurus that was then populated with the support of the neural network. For both datasets we obtained *Precision* and *Recall* equal to 100% after adding new synonyms to the thesaurus to resolve all the cases not perfectly matched.

	Relevant	Not-relevant	Total
Matched	866	0	866
Not-matched	0	449	770
Total	866	449	1315
<p>Recall 100%</p> <p>Precision 100%</p> <p>$F_{\alpha=0.5}$ 100%</p> <p>OA 100%</p>			

(a)

	Relevant	Not-relevant	Total
Matched	755	0	755
Not-matched	111	449	560
Total	866	449	1315
<p>Recall 87.18%</p> <p>Precision 100%</p> <p>$F_{\alpha=0.5}$ 93.15%</p> <p>OA 91.56%</p>			

(b)

Table 2. Evaluation metrics for the problem 'cell-phone' with thesaurus (a) and without thesaurus(b).

	Relevant	Not-relevant	Total
Matched	1054	0	1054
Not-matched	0	1658	1659
Total	1054	1658	2712
Recall 100%			
Precision 100%			
$F_{\alpha=0.5}$ 100%			
OA 100%			

(a)

	Relevant	Not-relevant	Total
Matched	963	0	963
Not-matched	91	1658	1749
Total	1054	1658	2712
Recall 91.37%			
Precision 100%			
$F_{\alpha=0.5}$ 95.49%			
OA 96.64%			

(b)

Table 3. Evaluation metrics for the problem 'digital-camera' with a thesaurus (a) and without thesaurus(b).

5 Conclusions and Future Works

The present work tested a system that make use of IE and classification in the context of a price comparison service. The approach proved highly accurate but it requires the assistance of an expert during the construction of the *KB* and the corresponding thesaurus.

Future work will extend the present solution including a tool for building the *KB* by automatically extracting unknown models of a product from a document.

References

1. Chang, C.H., Hsu, C.N., Lui, S.C.: Automatic information extraction from semi-structured web pages by pattern discovery. *Decis. Support Syst.* **35**(1) (April 2003) 129–147
2. Muslea, I.: Extraction patterns for information extraction tasks: A survey. In Califf, M.E., ed.: *Papers from the Sixteenth National Conference on Artificial Intelligence (AAAI-99) Workshop on Machine Learning for Information Extraction*, Orlando, FL, AAAI Press (July 1999)
3. Chang, C.H., Kayed, M., Girgis, M.R., Shaalan, K.F.: A survey of web information extraction systems. *IEEE Transactions on Knowledge and Data Engineering* **18**(10) (2006) 1411–1428
4. Damerau, F.J.: A technique for computer detection and correction of spelling errors. *Communications of the Association for Computing Machinery* **7**(3) (1964) 171–176
5. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation. (1986) 318–362
6. Jackson, P., Moulinier, I.: *Natural Language Processing for Online Applications: Text Retrieval, Extraction, and Categorization (Natural Language Processing, 5)*. John Benjamins Publishing Co (June 2002)