A Linguistic Inspection of Textual Entailment

Maria Teresa Pazienza[1], Marco Pennacchiotti[1], and Fabio Massimo Zanzotto[2]

[1] University of Roma Tor Vergata, Via del Politecnico 1, Roma, Italy,
{pazienza, pennacchiotti}@info.uniroma2.it,
[2] DISCo, University of Milano Bicocca, Via B. Arcimboldi 8, Milano, Italy,
zanzotto@disco.unimib.it

Abstract. Recognition of textual entailment is not an easy task. In fact, early experimental evidences in [1] seems to demonstrate that even human judges often fail in reaching an agreement on the existence of entailment relation between two expressions. We aim to contribute to the theoretical and practical setting of textual entailment, through both a linguistic inspection of the textual entailment phenomenon and the description of a new promising approach to recognition, as implemented in the system we proposed at the RTE competition [2].

1 Introduction

Several applications like Question Answering (QA) and Information Extraction (IE) strongly rely on the identification in texts of fragments answering specific user information needs. For example, given the question : *Who bought Overture?* a QA system should be able to extract and return to the user forms like *Yahoo bought Overture*, *Yahoo owns Overture*, *Overture acquisition by Yahoo*, all of them conveying equivalent or inferable meaning. A huge amount of linguistic and semantic knowledge is needed in order to find equivalences and similarities both at lexical and syntactic levels. Both the study of syntactic alternation and normalization phenomena [3], and the use of semantic and lexical resources, such as WordNet [4], could be useful to disentangle the problem from a linguistic perspective. On the contrary, most applications adopt statistical approaches, looking at collocation and co-occurrence evidences, avoiding deeper and complex linguistic analysis.

Whatever the approach is, what still lacks in the NLP community (as underlined in [3]) is the identification of a common framework in which to analyse, compare and evaluate these different techniques. Indeed, even if shared test beds and common roots for specific applications already exist (e.g., TREC competition for QA [5] and MUC [6] for IE), there is an emerging need for gathering together researches and methodologies that share the underlying common goal of equivalence/similarity recognition among surface forms.

In this direction, the *textual entailment* task has been recently proposed in [3] as a new framework, whose aim is to capture the common *core* shared by most NLP applications.

Roughly, textual entailment can be defined as an oriented relation, between the text T and the hypothesis H. T is said to entail H if the meaning of H can be inferred from the meaning of T. For example the sentence *Yahoo bought Overture* entails *Overture acquisition by Yahoo*. A system able to recognize textual entailment relations could be thus intended as the core of any NLP architecture whose aim is to extract an information need from textual material, looking at equivalence and more sophisticated subsumption phenomena.

Recognition of textual entailment is not an easy task. In fact, early experimental evidences in [1] seems to demonstrate that even human judges often fail in reaching an agreement on the existence of entailment relation between two expressions. In [1] it is shown that three judges could not reach an agreement in 233 over 759 cases of entailment template forms like T:[X punish Y] *entails* H:[X conquers Y]. Despite the intrinsic difficulties of the task, the recent first *Recognition of Textual Entailment* (RTE) competition [2] has seen the participation of a variety of systems, whose performance have been evaluated over a common test bed of T-H entailment pairs. The average accuracy performance of the proposed systems is still less than 60%.

We aim to contribute to the theoretical and practical setting of textual entailment, through both a linguistic inspection of the textual entailment phenomenon and the description of a new promising approach to recognition, as implemented in the system we proposed at the RTE competition [7].

2 Textual Entailment

2.1 Definitions

Textual Entailment is formally defined [3] as a relationship between a coherent text T and a language expression, the hypothesis H. T is said to entail H ($T \Rightarrow$H) if the meaning of H can be inferred from the meaning of T. An *entailment function* $\mathcal{E}(T, H)$ thus maps an *entailment pair* $T - H$ to a true value (i.e., *true* if the relationship holds, $false$ otherwise). Alternatively, $\mathcal{E}(T, H)$ can be also intended as a probabilistic function mapping the pair $T - H$ to a real value between 0 and 1, expressing the confidence with which a human judge or an automatic system estimate the relationship to hold.

In this perspective, two types of entailment can be identified:

- *Paraphrasing*: T and H carry the same fact, expressed with different words. For example *Yahoo acquired Overture* and *Yahoo bought Overture*.
- *Strict entailment*: T and H carry different facts, where the latter can be inferred from the former, as it is the case for *Yahoo bought Overture* and *Yahoo owns Overture*.

In textual entailment, the only restriction on T and H is to be meaningful and coherent linguistic expressions: simple text fragments, such a noun phrase or single words, or complete sentences. In the first case, entailment can be verified simply looking at synonymy or subsumption relation among words. For example

the entailment *cat* $\Rightarrow$ *animal* holds, since the meaning of the hypothesis (*a cat exists*) can be inferred from the meaning of the text (*an animal exists*). In the latter case, deeper linguistic analysis are required, as sentential expression express complex facts about the world. In this paper we will focus on sentential entailment, as it reveals the most challenging and practical issues.

In view of using entailment recognition methods in real NLP applications, two main tasks can be identified: entailment pattern *acquisition* and entailment *recognition*. Entailment patterns, as defined in [3], are formed by a T template and an H template, that is, two language expressions accompanied with syntactic properties and possible free slots. For example, the pattern $X_{subj} : buys : Y_{obj} \rightarrow X_{subj} : owns : Y_{obj}$ states that for any syntactically coherent instantiation of X and Y, entailment holds.

Entailment pattern acquisition aims at collecting these generalized forms, carefully inspecting textual corpora, using different techniques ranging from statistical counts to linguistic analysis [1, 8, 9]. Then they will be used to retrieve entailment relations in new texts, as needed for specific applications.

Entailment recognition aims at verifying if an entailment relation holds (possibly with a degree of confidence) between two linguistic expressions. Unlike patterns acquisition, in the recognition task the textual material of T and H is given. The use of linguistic resources and analysis can be thus preferred [10].

2.2 Type of Entailment

From an operational point of view, three types of entailment can be defined:

- *semantic subsumption*. T and H express the same fact, but the situation described in T is more specific than the situation in H. The specificity of T is expressed through one or more semantic operations. For example in the sentential pair H:[the cat eats the mouse], T:[the cat devours the mouse], T is more specific than H, as *eat* is a semantic generalization of *devour*.
- *syntactic subsumption*. T and H express the same fact, but the situation described in T is more specific than the situation in H. The specificity of T is expressed through one or more syntactic operations. For example in the pair H:[the cat eats the mouse], T:[the cat eats the mouse in the garden], T contains a specializing prepositional phrase.
- *direct implication*. H expresses a fact that is implied by a fact in T. For example H:[The cat killed the mouse] is implied by T:[the cat devours the mouse], as it is supposed that *killed* is a precondition for *devour*. [1]

Despite the two types of subsumption entailment, *direct implication* underlies deeper semantic and discourse analysis. In most cases, as implication concerns two distinct facts in T and H, and as facts are usually expressed through verbs, it follows that the implication phenomenon is strictly tied to the relationship among the T and H verbs. In particular, it is interesting to notice the temporal

[1] In [3] syntactic subsumption roughly corresponds to the *restrictive extension* rule, while direct implication and semantic subsumption to the *axiom rule*

relation between T and H verbs, as described in [4]. The two verbs are said to be in *temporal inclusion* when the action of one verb is temporally included in the action of the other (e.g. *snore→sleep*). *Backward-presuppoasition* stands when the H verb happens before the T verb (*win* entails *play*). Finally, in *causation* a stative verb in H necessarely follows a verb of change in T (e.g. *give→have*). In this case, the temporal relation is thus inverted with respect to backward-presupposition. Such considerations leave space to the application of temporal and verb analysis techniques both in the acquisition and recognition tasks.

3 Modelling Textual Entailment as Syntactic Graph Similarity

In this section we introduce the model of the running system for entailment *recognition* we presented at the RTE competition [7], based on the idea that entailment recognition can be modelled as a graph matching problem. Given a set of pairs $T - H$, the task consists in predicting if entailment holds or not, possibly accompanying the prediction with a confidence score.

As textual entailment is mainly concerned with syntactic aspects (as outlined in [3]), a model aiming at entailment recognition should basically rely on lexical and syntactic techniques, accompanied only with shallow semantic analysis.

We decided to model the entailment pair $T - H$ as two *syntactic graphs* augmented with lexical and shallow semantic information. Each graph *node* represents a phrase of the sentence, accompanied by its syntactic, morphological and semantic information. Each graph *edge* expresses a syntactic relation among phrases. Recognition can be thus intended as a process of comparison between two graphs: the more similar the graphs are, higher it is the probability of entailment. A *similarity measure* among graphs can be intended as a measure of entailment, and *graph matching theory* can be used as a tool to verify if entailment relation holds.

3.1 Graph Matching and XDG Basic Concepts

Graph matching theory aims at evaluating the similarity between two graphs. The power of graph matching is in the generality of graphs, as they can be used to represent roughly any kind of objects. Graph nodes usually represent object parts, while edges represent relations among parts. Matching algorithms are then used to recognize how similar two object are, looking at the structural similarity of their graph representation. Basic definitions of graph matching theory are those reported in [11]:

Definition 1. *A* graph *is defined as tuple $G = (N, E, \mu, \nu)$, where N is the finite set of labelled nodes, E the finite set of labelled edges connecting the nodes in N, $\mu : N \rightarrow L_N$ the function that assigns labels to nodes, and $\nu : E \rightarrow L_E$ the function that assigns labels to edges.*

Definition 2. *A* graph isomorphism *is a bijective function $f : N \rightarrow N'$, from a graph $G = (N, E, \mu, \nu)$ to a graph $G' = (N', E', \mu', \nu')$, such that:*

- $\mu(n) = \mu'(f(n))$ *for all* $n \in N$
- *for any edge* $e \in E$ *connecting two nodes* n_1, n_2, *it exists a edge* e' *connecting* $f(n_1), f(n_2)$, *and vice versa.*

Definition 3. *A* subgraph isomorphism *is an injective function* $f : N \to N'$, *from* $G = (N, E, \mu, \nu)$ *to* $G' = (N', E', \mu', \nu')$, *if it exists a subgraph* $S \subseteq G'$ *such that* f *is a graph isomorphism from* G *to* S.

Definition 4. *A graph* G *is called* common subgraph *between two graphs* G_1 *and* G_2 *if it exist a subgraph isomorphism from* G_1 *to* G *and from* G_2 *to* G.

Definition 5. *The common subgraph* G *of* G_1 *and* G_2 *with the highest number of nodes is called the* maximal common subgraph *(*$mcs(G_1, G_2)$*).*

The concept of *maximal common subgraph* (*mcs*) is often central in the definition of a *similarity measure*. In fact, in real applications errors and distortions in the input graphs can easily appear. Consequently, as perfect matching between two object is often impossible, graph matching algorithms must be error tolerant, returning as result a degree of similarity between graphs, rather than a deterministic matching answer.

In the context of textual entailment, graph matching theory must be applied to two graphs representing the syntactic structure of T and H, together with relevant lexical information. As useful syntactic representation we decided to use the extended dependency graph (XDG) [12]. An XDG is basically a dependency graph whose nodes C are *constituents* and whose edges D are the *grammatical relations* among the constituents, i.e. $\mathcal{XDG} = (C, D)$. Constituents, i.e. $c \in C$, are classical syntactic trees with explicit *syntactic heads*, i.e. $h(c)$, and *potential semantic governors*, i.e. $gov(c)$. A constituent can be either *complex* or *simple*. A *complex constituent* is a tree containing other constituents as sons. A *simple constituent* represent a leaf node, i.e., a token span in the input sentence, that carries information about lexical items (surface form, lemma, morphology, etc.). Dependencies in $(h, m, T) \in D$ represent typed (where T is the type) and ambiguous relations among a constituent, the *head* h, and one of its *modifiers* m. Ambiguity is represented using *plausibility* (between 0 and 1).

The syntactic analysis of entailment pairs has been carried out by Chaos [12], a robust modular parser based on the XDG formalism.

3.2 Adapting XDG and Graph Matching to Textual Entailment

Entailment recognition can thus be intended as a matching process among the two XDG graphs representing the hypothesis and the text, where nodes are the set of constituents C and edges are the set of dependencies D.

Concerning the XDG formalism, it is necessary to define the specific kind of information that a graph devoted to entailment recognition must hold. Then, it must be verified if XDG graphs are able to capture all these information. The three type of entailment outlined in Sec. 2.2 require, in order to be detected:

– *syntactic information*. In general, graphs that have similar syntactic and surface structure are likely to express the same fact. Moreover, syntactic addition to the T graph with respect to H can reveal a strict entailment relation, as capturing *syntactic subsumption entailment*. Finally, syntactic variations such as nominalization and active/passive transformations must be treated as invariant operations on graphs.
– *shallow lexical-semantic information*. Syntactic similarity can be supported by lexical-semantic information needed to grasp *semantic subsumption entailment*, such as verb and noun generalization, antinomy and synonymy. Moreover, *direct implication* requires the recognition of verb entailments.

The XDG formalism captures all needed information, as syntactic dependencies are explicitly represented, and lexical information about nodes are carefully treated.

A classical graph matching problem and textual entailment present similarities:

– In both cases there are complex objects to be matched.
– In order to tackle errors and distortion, in both tasks it is preferred to adopt a similarity measure able to express the degree of similarity between two object (e.g. using *mcs*), rather than a deterministic value.

However, some peculiar properties of textual entailment make necessary major adaptations of the standard graph matching methodology:

– *Node complexity*. In the graph theory nodes are matched simply looking at their *label level*. In textual entailment node similarity can not be reduced to a surface analysis, as both morphological and semantic variations must be taken into account. Textual entailment nodes are not atomic, since they represent complex constituents that can be further divided in sub-constituents for deeper lexical-semantic analysis. For these two reasons, matching between two nodes is a complex process. It is necessary to evaluate a graded level of linguistically motivated *node semantic similarity* $sm(c_h, c_t)$.
– *Edge complexity*. Edges are complex structures too: their matching must be evaluated looking also at the type of dependency they express. A graded *syntactic similarity* $ss(c_h, c_t)$ has then to be defined to capture this aspects.
– *Transformation invariance*. Textual entailment must account for graph invariant transformations: specific type of syntactic phenomena (nominalization, active/passive transformation, etc.) should be properly treated. Two graphs representing syntactic variations of the same fact, while structurally dissimilar, should be considered as equivalent.
– *Asymmetry*. Textual entailment, unlike the classical graph problems, is not symmetric, since it represents a direct relation of subsumptiom from T to H. By consequence, the *graph isomorphism* definition must be further refined in a more specific notion of *XDG subsumption isomorphism*.

In view of these observations, definition in Sec.3.1 are extended as follows.

Definition 6. *An* XDG subsumption isomorphism *is an oriented relation from a text* $\mathcal{XDG}_T = (C_T, D_T)$ *to an hypothesis* $\mathcal{XDG}_\mathcal{H} = (C_H, D_H)$ *(*$\mathcal{XDG}_H \preceq \mathcal{XDG}_T$*), expressed by two bijective functions:*

- $f_C : C_T \rightarrow C_H$
- $f_D : D_T \rightarrow D_H$

where f_C *and* f_D *describe the oriented relation of subsumption between constituents (nodes) and dependencies (edges) of* H *and* T*.*

f_C and f_D play the role of function f in the definition of *graph isomorphism* in Sec.3.1. Unluckily, due to the *node and edge complexity* factors, a definition of f_C and f_D can not be easily stated as for f. Sec.3.3 will thus give an extensive description on how these two functions are modelled.

Definition 7. *A* subgraph subsumption isomorphism *between* $\mathcal{XDG}_H$ *and* $\mathcal{XDG}_T$*, written as* $\mathcal{XDG}_H \sqsubseteq \mathcal{XDG}_T$*, holds if it exists* $\mathcal{XDG}'_T \subseteq \mathcal{XDG}_T$ *so that* $\mathcal{XDG}_H \preceq \mathcal{XDG}'_T$*.*

Isomorphic subsumption aims to capture cases 1 and 3 described in Sec.2.2, while *subgraph subsumption isomorphism* corresponds to case 2 in Sec.2.2.

As in graph matching theory, an *mcs* must be defined in order to cope with distortions and errors in the input graphs mainly introduced by syntactic parser erroneous interpretations.

Definition 8. *The* maximal common subsumer subgraph *(*mcss*) between* $\mathcal{XDG}_H$ *and* $\mathcal{XDG}_T$ *is the graph with the highest number of nodes, among all the subgraph of* $\mathcal{XDG}_H$ *which are in isomorphic subgraph subsumption with* $\mathcal{XDG}_T$*.*

3.3 Graph Syntactic Similarity Measure for Textual Entailment

The similarity measure $\mathcal{E}(\mathcal{XDG}_T, \mathcal{XDG}_H)$, used to estimate the degree of confidence with which $\mathcal{XDG}_H$ and $\mathcal{XDG}_T$ are in entailment relation, must be modelled on the subsumption between nodes and edges in T and H, grasping the notion of *mcss*. Four main steps are required:

1. *Model the bijective function* $f_C : C'_T \rightarrow C'_H$, that maps constituents in $C'_H \subseteq C_H$ to subsuming constituents in $C'_T \subseteq C_T$. A semantic similarity $sm(c_h, c_t)$ must be accompanied to each mapping. For example in the pair H:[the cat eats the mouse], T:[the cat devours the mouse], *eats* could be mapped in *devours*.
2. *Model the bijective function* $f_D : D'_T \rightarrow D'_H$, that maps dependencies in $D'_H \subseteq D_H$ to dependencies in $D'_T \subseteq D_T$. A syntactic similarity $ss(c_h, c_t)$ is then derived to better capture the implications of such mappings.
3. *Find the mcss*, that is, the common subgraph identified by f_C and f_D. The *mcss* must be accompanied with an overall similarity, deriving from the *sm* and *ss* of its nodes and edges.
4. *Model* $\mathcal{E}(XDG_T, XDG_H)$ using *mcss* and the two input graphs $\mathcal{XDG}_H$ and $\mathcal{XDG}_T$. Textual entailment between a pair $T - H$ will be thus predicted verifying $\mathcal{E}(XDG_T, XDG_H)$ against a manually tuned threshold.

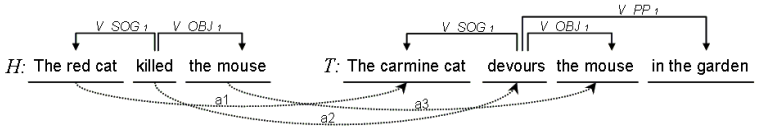

Fig. 1. A complete example of entailment pair, represented in the XDG formalism. Solid lines indicate grammatical relations D (with *type* and *plausibility)*; dotted lines indicate anchors a_i between H and T constituents.

Node Subsumption The node subsumption function f_C must identify constituents in C_H that can be mapped to constituents C_T. We will define the function with a set A containing the *anchors*, i.e. the correspondences between the constituents of C_H and C_T. The set A will thus represent the nodes of the *mcss*.

In A, each constituent $c_h \in C_H$ is associated, if possible, to its most similar constituent $c_t \in C_T$ (that is, the c_t that most likely subsumes c_h). The definition follows:

Definition 9. *Given the* anchors *$a = (c_h, c_t)$ as linking structures, connecting constituents $c_h \in C_H$ to constituents $c_t \in C_T$ and a function of* semantic similarity *$sm(c_h, c_t) \in (0, 1]$ expressing how much similar c_h and c_t are looking at their lexical and semantic properties, the set of anchors A is:*

$$A = \{(c_h, c_t) | c_h \in C_H, c_t \in C_T, sm(c_h, c_t) = \max_{c \in C_T} sm(c_h, c) \neq 0\}$$

If a subsuming c_t can not be found for a c_h (i.e. $\max_{c \in C_T} sm(c_h, c) = 0$), then c_h has no anchors. For example in the entailment pair of Fig. 3.3, f_C produces the mapping pairs *[The red cat - The carmine cat], [killed - devours], [the mouse - the mouse].*

The semantic similarity $sm(c_h, c_t)$ is derived on the basis of the syntactic type of c_h, that is, if it is a noun-prepositional phrase $sm(c_h, c_t) = sm_{np}(c_h, c_t)$ or a verb phrase $sm(c_h, c_t) = sm_{vp}(c_h, c_t)$. If c_h is a noun-prepositional phrase, similarity $sm_{np}(c_h, c_t)$ is evaluated as:

$$sm_{np}(c_h, c_t) = \alpha * s(gov(c_h), gov(c_t)) + (1 - \alpha) * \frac{\sum_{sh \in S(c_h)} \max_{s_t \in S(c_t)} s(s_h, s_t)}{|S(c_h)|}$$

where $gov(c)$ is the governor of the constituent c, $S(c_h)$ and $S(c_t)$ are the set simple constituents excluding the governors respectively of c_h and c_t, and $\alpha \in [0, 1]$ is an empirically evaluated parameter used to weight the importance of the governor. In turns, $s(s_h, s_t) \in [0, 1]$ expresses the similarity among two simple constituents: it is maximal if they have same surface or stem (e.g. *cat* and *cats*), otherwise a semantic similarity weight $\beta \in (0, 1)$ is assigned looking at possible WordNet relations (synonymy, entailment and generalization).

If c_h is a verb phrase, different *levels* of similarity are taken into consideration, according to the semantic value of its modal. For example *must go-could go*

should get a lower similarity than *must go-should go*. A verb phrase is thus composed by its governor *gov* and its modal constituents *mod*. The overall similarity is thus:

$$sm_{vp}(c_h, c_t) = \gamma * s(gov(c_h), gov(c_t)) + (1 - \gamma) * d(mod(c_h), mod(c_t))$$

where $d(mod(c_h), mod(c_t)) \in [0, 1]$ is empirically derived as the semantic distance between two modals (e.g., *must* is nearer to *should* than to *could*) (classified as generic auxiliaries, auxiliaries of possibility and auxiliaries of obligation).

Edge Subsumption Once f_C is defined the existence of the bijective fuction f_D can be easly verified by construction. The edge subsumption function f_D maps $(c_h, c'_h, T_h) \in D_H$ to $f_D(c_h, c'_h, T_h) = (c_t, c'_t, T_t) \in D_T$ if $T_h = T_t$ and $(c_h, c_t), (c'_h, c'_t) \in A$. The set of mapped D_H will thus represent the edges linking the nodes of the *mcss*.

The definition of f_D gives the possibility of investigating the external *syntactic similarity* $ss(c_h, c_t)$ of a given anchor $(c_h, c_t) \in A$. This should capture how similar are the relations established by elements in the anchor. Our syntactic similarity $ss(c_h, c_t)$ depends on the semantic similarity of the constituents connected with the same depencency to c_h and c_t in their respective $\mathcal{XDG}$s, that is, the set $A(c_h, c_t)$ defined as:

$$A(c_h, c_t) = \{(c'_h, c'_t) \in A | f_D(c_h, c'_h, T) = (c_t, c'_t, T)\}$$

For example in Fig. 3.3, $A(killed, devours)$= {([*the_red_cat*], [*the_carmine_cat*]), ([*the_mouse*], [*the_mouse*])}. The syntactic similarity $ss(c_h, c_t)$ is then defined as:

$$ss(c_h, c_t) = \frac{\sum_{(c'_h, c'_t) \in A(c_h, c_t)} sm(c'_h, c'_t)}{|D_H(c_h)|}$$

where $D_H(c_h)$ are the dependencies in D_H originating in c_h.

Similarity measure Once nodes and edges of the *mcss* have been identified through f_C and f_D, an overall similarity $S(mcss)$ is evaluated for *mcss*. $S(mcss)$ must express how much similar the two subgraphs $\mathcal{XDG}'_T$ and $\mathcal{XDG}'_H$ in isomorphic subsumption are, both from a syntactic and a semantic point of view.

For each pair $(c_h, c_t) \in A$ a global similarity S is thus derived as:

$$S(c_h, c_t) = \delta * sm(c_h, c_t) + (1 - \delta) * ss(c_h, c_t)$$

where δ is a manually tuned parameter. The, similarity measure $\mathcal{E}(\mathcal{XDG}_T, \mathcal{XDG}_H)$ can be evaluated in analogy to the measure described in 3.1. In this specific case, numerator and denominator will not be expressed as number of nodes, but as probabilities, since, as stated before, textual entailment must account for node and edges complexity. Numerator will thus be the overall *mcss* similarity . The denominator will express the best case, in which *mcss* corresponds to $\mathcal{XDG}_{\mathcal{H}}$, and all nodes and edges match with probability 1 to elements of a hypothetic T.

$$\mathcal{E}(\mathcal{XDG}_T, \mathcal{XDG}_H) = \frac{S(mcss)}{|C_H|} = \frac{\sum_{(c_h,c_t)\in A} S(c_h, c_t)}{|C_H|}$$

3.4 Graph Invariant Transformations

Entailment pairs are often expressed through syntactic variations, as:

H:[The cat killed the mouse], T:[The killing of the mouse by the cat]

We had thus to model the most important variation phenomena in our system, in order to cope with pairs with different syntactic structures used that express the same fact. Before the graph matching procedure, a set of graph transformation rules have been applied to $\mathcal{XDG}_H$ and $\mathcal{XDG}_T$, in order to bring to a normalized form sentences that have a syntactic variation. For example in the abovementioned example, the text is brought back to the normal form T:[the cat killed the mouse]. We modelled the following type of invariant transformation:

- *nominalization in T*. Different cases such as T:[The killing of the mouse by the cat] and T:[The cat is the killer of the mouse] are treated. Only nominalization of T is taken into consideration, as usually in entailment relations nominalization happens only in T;
- *passivization in H or T*. Passive sentences are brought to active forms.
- *negation in H or T*. If one sentence is the negative form of the other, the two sentences are recognized to be not in entailment (*negative subsumption*).

4 Experimental Setting and Result Analysis

The RTE challenge has been the first test to verify the performances of our system. The data set used for the competition was formed by three sets of entailment pairs. A First development set, composed by 287 annotated pairs; a *Second development set*, composed by 280 annotated pairs; a *Test set*, composed by 800 non annotated pairs. In the first two sets the true value of entailment was given, in order to model and tune the systems participating at the competition. The test set was used for evaluation. We used the first development set to tune parameters of the model (α, β, γ, δ) and the second development set to verify the validity of the model, while the test set was used for the competition. Participating systems were evaluated over the test set: a prediction value ($True/False$) and an associated degree of confidence on the prediction $c \in [0, 1]$ have been provided for each pair. Two measures were used for evaluation: *accuracy* (fraction of correct responses) and the *confidence-weighted score (cws)* as defined in [2].

Performance of our system where on the average of those reported by other participants (cws 55,70%). Results are shown on Table 1 . In fact, pairs of development and test sets have been collected by human annotators, looking at typical NLP *tasks*: Information Retrieval (IR), Information Extraction (IE), Comparable Documents (CD), Reading Comprehension (RC), Question Answering (QA),

Measure	*Result*
cws	0.5574
accuracy	0.5245
precision	0.5265
recall	0.4975
f	0.5116

TASK	*cws*	*accuracy*
CD	0.8381	0.7651
IE	0.4559	0.4667
MT	0.5914	0.5210
QA	0.4408	0.3953
RC	0.5167	0.4857
PP	0.5583	0.5400
IR	0.4405	0.4444

Table 1. RTE competition results on data set (800 entailment pairs). On the left, overall results. On the right, results ordered by task.

Machine Translation (MT), Paraphrase Acquisition (PP). As different application envision different types of linguistic expression and consequently different types of entailment, it is predictable to obtain different performances for pairs collected in different ways.

At a first glance, results in Table 1 seems to show low accuracy and cws. In truth, performances are in line with those obtained by other systems, driving at least to two conclusions. Firstly, the task of entailment recognition seems inherently hard, as outlined in Sec.1. Secondly, low results indicate that research on entailment is only at an early stage and that system proposed must be considered as prototypical architectures implementing on-going studies.

More in particular, it is interesting to analyze results looking at the performance obtained on the different tasks. As Table 1 shows, our system performed quite well on the CD task, while worst performance were achieved in QA and IR. Such large difference is not surprising, as a comparison of the pairs of each tasks reveals. CD pairs are characterized by having T and H syntactically and lexically very similar, e.g.:

T: [The first settlements on the site of Jakarta were established at the mouth of the Ciliwung, perhaps as early as the 5th century AD.]

H: [The first settlements on the site of Jakarta were established as early as the 5th century AD.]

IR and QA pairs are much more complex to recognize, as they are usually formed by a brief H statement and a long an complex T sentence, e.g.:

T: [A former police officer was charged with murder Thursday in the slaying of a college student who allegedly had threatened to expose their sexual relationship, authorities said.]

H: [Ex-cop Rios killed student.]

As the graph method we adopted is strongly based on the comparison of the syntactic graphs of T and H, it follows that it is most suited to the CD task, as the system can easily find anchors and carry out the reasoning. On the contrary, for IR and QA task statistical approaches to entailment recognition would maybe be the only strategy to approach the task.

In general, as expected, the system has been able to successfully recognize entailment when the relation between T and H was expressed through syntac-

tically *recognizable* operations (e.g., invariant transformations and constituent generalizations/synonymy):

T: [Korean Seoul, formally Soul-t'ukpyolsi ("Special City of Seoul"), largest city and capital of the Republic of Korea (South Korea), ...]

H: [The capital and largest city of South Korea is Seoul.]

5 Conclusions and Future Works

While a very important results must be considered the overall methodological definition of the task and the theoretical defintions given in Sec.2, early experimental results show that further improvement are needed, in particular to cover entailment cases that can be grasped only by statistical techniques. Future works thus envision a deeper analysis of entailment pairs in the perspective of the classifications proposed, in order to identify specific entailment patterns that could be associated to each specific entailment type. Moreover, the fertile combination of statistical and linguistic methods for both recognition and acquisition could be a promising area of investigation.

References

1. Szpektor, I., Tanev, H., Dagan, I., Coppola, B.: Scaling web-based acquisition of entailment relations. In Lin, D., Wu, D., eds.: Proceedings of EMNLP 2004, Barcelona, Spain, Association for Computational Linguistics (2004) 41–48
2. Dagan, I., Glickman, O., Magnini, B.: The pascal recognising textual entailment challenge. In: PASCAL Challenges Workshop, Southampton, U.K (2005)
3. Dagan, I., Glickman, O.: Probabilistic textual entailment: Generic applied modeling of language variability. In: Learning Methods for Text Understanding and Mining, Grenoble, France (2004)
4. Miller, G., Beckwith, R., Fellbaum, C., Gross, D., Miller, K.: Five papers on wordnet. Technical Report CSL Report 43, Princeton University (1990)
5. Voorhees, E.M.: Overview of the trec 2003 question answering track. In: TREC. (2003) 54–68
6. Proceedings of the Seventh Message Understanding Conference (MUC-7), Virginia USA, Morgan Kaufmann (1998)
7. Pazienza, M.T., Pennacchiotti, M., Zanzotto, F.M.: Textual entailment as syntactic graph distance: a rule based and a svm based approach. In: PASCAL Challenges Workshop, Southampton, U.K (2005)
8. Barzilay, R., McKeown, K.: Extracting paraphrases from a parallel corpus. In: Proceedings of the 39th ACL Meeting, Toulouse, France (2001)
9. Lin, D., Pantel, P.: DIRT,discovery of inference rules from text. In: Knowledge Discovery and Data Mining. (2001) 323–328
10. Hagege, C., Roux, C.: Entre syntaxe et smantique : normalisation de la sortie de l'analyse syntaxique en vue de l'amlioration de l'extraction d'information a partir de textes. In: TANL 2003, Batz-sur-Mer,France (2003)
11. Bunke, H.: Graph matching: Theoretical foundations, algorithms, and applications. In: Vision Interface 2000, Montreal, Springer-Verlag (2000) 82–88
12. Basili, R., Zanzotto, F.M.: Parsing engineering and empirical robustness. Natural Language Engineering **8/2-3** (2002)