

Towards a Black Box Approximation to Human Processing of Narratives Based on Heuristics over Surface Form

Carlos León¹ and Pablo Gervás²

¹ cleon@fdi.ucm.es

Departamento de Ingeniería del Software e Inteligencia Artificial

² pgervas@sip.ucm.es

Instituto de Tecnología del Conocimiento
Universidad Complutense de Madrid
Spain

Abstract

Computational Narrative has provided several examples of how to process narrations using semantical approaches. While many useful concepts for computational management of stories have been unveiled, a common barrier has hindered their development: semantic knowledge is still too complex to handle. In this paper, a focus shift based on narrative structure is proposed. Instead of digging deeper into the possibilities of semantic processing, analysing *structural* properties of stories and keeping the semantic load to a minimum can allow for a more efficient use of available narrative corpora, even without mimicking human behaviour.

Introduction

Semantical processing of narrations tries to reach, from one or another point of view, human understanding or generation of stories. This has been the predominant tendency in automatic story generation systems and other studies in the field (Meehan 1976; Turner 1992; Pérez y Pérez 1999; Bringsjord and Ferrucci 1999; Riedl 2004). While nothing prevents from success when reproducing the way in which humans perform narrative generation or understanding, current results evidence that this is not an easy task. Problems affecting Artificial Intelligence in general (knowledge acquisition bottleneck, efficiency for complex domains, and others) also appear in Computational Narrative.

These drawbacks seem to block the computational implementation of concepts from modern narratology (Herman 2000) and other disciplines like Psychology (Kelly 1955), which conceive narrations as cognitive processes not totally describable in terms of structural properties of stories, as opposed to the structuralist point of view of classic narratology (Propp 1928; Barthes and Duisit 1975). However, current computational techniques are still far from being able to model the inner processes that govern human understanding of stories. Realizing advances in narratology as computer programs is not directly feasible because narratology assumes conceptual capabilities exceeding those currently available in machines.

Nevertheless, much work has been done in the field of cognitive approaches to computational narrative, although

the previously introduced limits have set important bounds on the final results. BRUTUS (Bringsjord and Ferrucci 1999), for instance, proposes a knowledge intensive system whose results show high quality stories, non-easily distinguishable from a human work. From a simplistic point of view it can be said that BRUTUS performs the generation based on a set of knowledge rules from different fields (narrative, psychological, linguistic) that collect human knowledge. These rules, however, are only able to generate a reduced set of stories. While these are relatively complex, the effort of identifying, encoding and testing these rules is probably much higher than the effort required to write the same amount of stories of similar quality by hand. Similar issues arise in other story generation systems with varying qualities and ratios between human effort and number of correctly processed (generated, in most cases) stories (Meehan 1976; Turner 1992; Riedl 2004; Dehn 1981).

It can be concluded, then, that the efficiency of creating semantic systems for story processing, so far, is quite low. Several years are usually needed to develop a new system. Therefore, while research on the field of story processing based on semantic is absolutely valuable, new perspectives to solve the problem could help to build a more efficient solution in terms of the required time to develop new, useful systems.

As an alternative, there exist already some approaches devoted to non-semantic processing of narrative content. For example, Chambers and Jurafsky present a statistical learning approach which tries to extract probabilities of sequences of events in their narrative context, with promising results (Chambers and Jurafsky 2008; 2009). Their proposal is restricted to the so-called *narrative-chains*, which, while useful, do not allow for a fine grain use of the learnt content. Additionally, only narrative schemas are addressed, but complete *stories* are not studied. This is discussed later.

Between these extremes to computational narrative, namely cognitive and structural approaches, this paper proposes a model in which the amount of management of cognitive structures is kept to a minimum. While this option gets far from the current conception of cognitive processes for narratives, our knowledge about the way in which human psychology processes narrations is still too reduced to be able to create a complete cognitive model.

In order to increase the efficiency in computational narrative systems, a structural relation defined in terms of heuristics or *rules of thumb* is used in the model as a computationally processable property of narrations. Since every aspect of this property is defined in terms of observable features (structural layout of a graph), the meaning of the facts is ignored by the process. This means that any number of stories can be processed without adding additional knowledge to the system in rules or whatever formalism. There is an important loss when discarding the semantic properties of stories, though, and this is discussed later. However, preliminary tests show interesting results.

A Story Generation System based on Semantic Knowledge

León and Gervás (León and Gervás 2010) propose a story generation system based on an evaluation function for narratives (León and Gervás 2010). The generation algorithm was designed as a *generate and test* pattern in which exhaustive exploration of a space of stories is carried out. The exploration returns those visited stories for which the evaluation function yields that the story is “good”.

The system controls the generation through the application of the evaluation function itself. This function receives a story and outputs a value in the real range $[-1, 1]$. As reference, very bad stories receive the value -1 and great stories are assigned the value 1 , a correct but not exceptional story receives the value 0 .

The evaluation function in this described story generation system returns, as overall quality, the average value of 13 variables: *interest*, *causality*, *compression*, *danger*, *love*, *tension*, *humanity*, *action*, *hypotheses*, *empathy*, *funny*, *emotion*, *chronology* and *overall quality*. The definition of these variables can be checked in the paper defining the generation system (León and Gervás 2010). The value of these variables also ranges in the interval $[-1, 1]$. The evaluation function sequentially traverses the events describing the story in order, and iteratively updates the value for each one of these variables until the last event in the story is evaluated.

Since, by design, the evaluation function was implemented using a knowledge intensive approach, rules for the definition of the domain have to be created. Considering that nowadays no computational system is able to replace human labour, this has to be done by hand.

In the way rules were created, the human programmer has to consider the meaning of every event in the story and update the values of the 13 variables depending on that meaning. While this is complicated by itself because any event can have several many possible interpretations, it has to be taken into account that the meaning of a event depends also on the context. For instance, in the Iliad, “Achilles killed Hector” affects the interpretation of the story in a different way than “Hector killed Achilles” would have affected it, even considering that these two events contain the same characters and the same verb.

Therefore, the development of the rule-base scales with major problems: adding a new event implies checking the whole set of rules to update all the events whose interpreta-

tion is affected by a context in which the new information is included.

From Semantic to Surface Form Processing

The previously introduced system, despite of its inherent limitations, suggests a possible change of focus. The ad-hoc definition of the variables yield promising empirical results, even being domain-specific. And the evaluative nature of the solution offers explicit analysis of certain features of narrations, some of them involving structural aspects of stories. Since the definition of the features, while intendedly cognitive, just modelled an approximation to evaluation of stories, the study of the system led to the hypothesis that empirical *rules of thumb* about properties of narrative could permit the processing of narratives.

That is, it was hypothesized that just by taking into account surface properties of formalized story plots, stories understandable by humans as such could be generated.

Studying the results collected from human evaluation in the semantic story generator, it was detected that there was a strong agreement between evaluators’ criteria in two variables: *causality* and *chronology*. *Causality* tried to represent the perception that everything happens because of a reason, and *chronology* rates the correct layout of events in time (for instance, that causes happen before their effects). Being a very simple domain easily interpretable by all evaluators, all of them rated the quality about these two variables with very few differences.

Additionally, there was a clear correlation between *causality*, *chronology* and *overall quality*: stories receiving *causality* and *chronology* rating above zero also received rates for *overall quality* above zero. Zero was set as the threshold between acceptable and non-acceptable stories.

This is a common psychological process when humans understand narrative (Herman 2000), so it was considered that correctness could be indirectly modelled by replicating these kind of understanding processed.

Preconditional Links

A heuristic was created to capture this: the *preconditional link*. Preconditional links try to describe the structural patterns in narrations that are involved in the human recognition of causality and chronology, ignoring the semantic content of these two concepts. The preconditional link is defined next:

In a story formalized as the sequence of events $\{e_1, \dots, e_n\}$, the events $\{e_i, \dots, e_j\}$ are *preconditionally linked* to e_k iff:

- they appear before e_k in the sequence $\{e_1, \dots, e_n\}$,
- and the directed graph resulting from the preconditional links for all events in the story converges to the last event in the story,

where $1 \leq i, j < k \leq n$.

This definition is obviously structural, that is, it only captures surface properties of stories according to a synthetic relation. As an example, if a story is composed by the

events $\{a, b, c, d, e\}$, valid preconditional links would appear in Figure 1, but not in Figure 2.

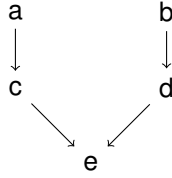


Figure 1: Valid set of preconditional links. The arrows represent preconditional links.

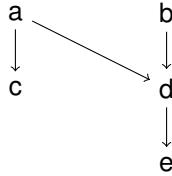


Figure 2: Non-valid set of preconditional links. The arrows represent preconditional links.

The definition of preconditional link is only inspired by the way in which, heuristically, humans perform story understanding, but it does not try to capture any cognitive process. Instead, the current proposal defines it as a heuristic for machines, and not for humans. That is, the definition is strictly bound to computationally processable information: not necessarily the information that humans use and not necessarily processed in the way humans do.

Black Box Definition of Story Processing

Based on the previous relation, a definition of story processing is proposed. In particular, story generation is addressed as an example of automatic processing. Although it has not been particularly studied, nothing prevents the definition of preconditional links from being applied to story evaluation, for instance.

The generation model is proposed as a black box approach from the human cognition point of view. Following this structural conception, it is assumed that humans must only consider stories as generated items without taking into account how the stories were generated. This is a main assumption for the model because humans, as it will be shown, are involved in the proposed execution model.

Figure 3 depicts a schematic representation of the execution model. In it, a *story* or a set of stories written by humans is translated into a machine processable *sequence* or to a set of sequences. How this translation must be carried out in a real scenario is not addressed in this paper from a theoretical point of view. The prototype implementation has been carried out by performing this translation by hand, which is acceptedly a non-general approach. While there are some tools that address this task (Klein and Manning 2003), the details of this process are considered to be outside the scope

of this paper. Future work contemplates improving this area of the model.

After the stories have been translated to formal sequences (Figure 4 shows an example), the sequence is processed by an *identification* process in which a valid set of *preconditional links* are found, thus obtaining a representation of the story similar to the one in Figures 1 and 2.

The proposed identification process has been implemented for the prototype as a generate and test algorithm. Candidate preconditional links in a story are generated and only those which satisfy the constraint of converging to a single event are considered valid sets of preconditional links for a story. In the implementation, the first valid candidate is chosen as output. It would make sense to test the whole approach by considering more than the first candidate, but this has not been addressed yet (it is planned as part of the future work).

After the preconditional links have been collected, they are abstracted in order to get *preconditional rules*. Preconditional rules are a simple abstraction in which the particular case $a \rightarrow b$, for instance, is turned to $\alpha \rightarrow \beta$, in which α and β are general cases of a and b . More specifically, if a is “John goes to the shop” and b is “Mary buys a car”, α would now be “someone goes to the shop” and β is “someone buys a car”. The proposed theoretical model does not impose any restrictions about how this abstractions must be done since it is considered that it is dependent on the particular implementation.

In the particular implementation that has been programmed for testing the model, events are encoded as tuples of verbs and arguments, in the form $love(john, mary)$ or $take(mary, glass)$, meaning that “John loves Mary” and “Mike takes the glass”, respectively. For a preconditional link of $love(john, mary) \rightarrow take(mary, glass)$, the abstraction process would generate rule in the form of $love(x?, y?) \rightarrow take(y?, z?)$, where $\{x?, y?, z?\}$ are variables which could be unified with terminal tokens in generation (these would be $john$, $mary$ or $glass$, for instance). It is important to note that the terminals in the preconditional links are translated into variables in rules according to their position in the links, following a simple variabilization process (Charniak and McDermott 1985): $john$ is translated to $x?$ for every appearance of $john$. This makes the rules keep the form of the links, which has been proven to be empirically useful in the example story generation.

After having gathered the set of preconditional rules through an abstraction process, these are used to perform simple story generation. The proposed model considers a rule-set partitioned in *good rules* and *bad rules*. The set of good rules contains those abstracted preconditional links present in stories validated as correct by humans (as shown below). The set of bad rules contains the abstracted preconditional links which are present in stories classified as non-correct *if* these abstracted preconditional links are not in the set of good rules. If a rule is considered “bad” at some stage, and then it is considered “good”, it is finally classified as “good”. That is, the validation of a rule as “good” can not be undone, but a “bad” rule can be considered good at some later point.

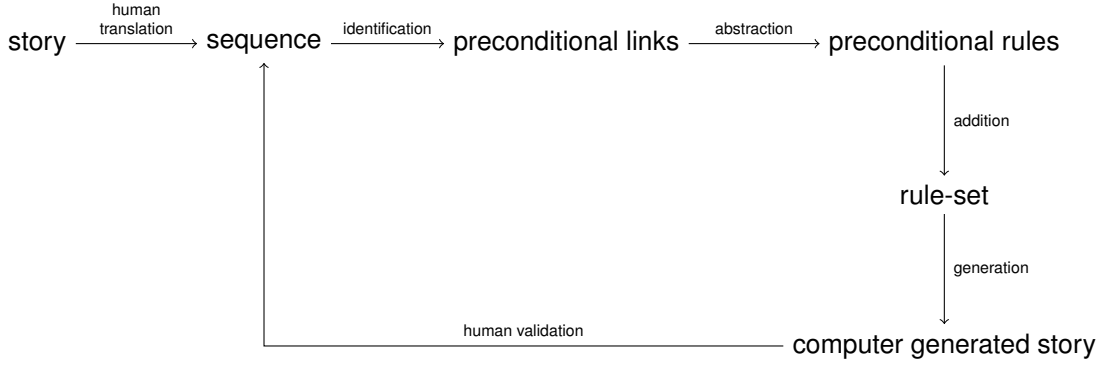


Figure 3: Process of structural processing of stories.

That is, if a story s_i was considered correct, and it led to the creation of the preconditional rules $\{r_1, r_2, r_3\}$ and a non-correct story s_j led to the creation of rules $\{r_1, r_4, r_5\}$, the resulting *good* subset would contain the rules $\{r_1, r_2, r_3\}$ and the *bad* subset would contain the rules $\{r_4, r_5\}$.

There are many more ways of creating the rule set, but for prototyping the system and for explanation purposes this approach is simple and useful. The study of more sophisticated ways will be addressed in further research.

Again, the particular way in which story generation can be performed is outside the scope of this paper, and many approaches could make use of this information. The simplistic algorithm that has been used for the preliminary evaluation of this model is based on a simple application of the rules.

For instance, if an execution considers the rules $\alpha \rightarrow \beta$, $\beta \rightarrow \gamma$ and $\gamma \rightarrow \delta$, valid instantiation of the variables (α as a , β as b , and so on) would generate the stories $\{a, b, c\}$ or $\{a, b, c, d\}$.

The set of *bad* rules is used to prevent the appearance of “wrong” preconditional links in the generation of stories. If a generated story contains a sequence of events which could be matched against a rule in the *bad* set, the story is discarded. This can happen even in a direct application of the *good* rules because many preconditional links can be inferred in a story according to the definition and the generate and test pattern used to find the preconditional links. In this way, both *good* and *bad* rules are used.

Human validation is used after the generation to check that the generated stories are correct. So far, the generated stories could be acceptable or not according to the evaluator’s criteria, but if the execution was concluded here, the limits of cognitive story processing would have not been addressed because this structural approach would generate only a limited amount of data, those coming from the input set of stories.

Therefore, the model is completed by a process that iteratively refines in a pseudo-automatic way the generative capabilities of the algorithm shown in Figure 3 by collecting preconditional rules at each step.

Acquisition of Preconditional Rules Through Human Validation

Following the schematic depiction of the structural processing of stories algorithm in Figure 3, it can be seen that there is a feedback process from the computer generated story to a new sequence. This feedback is proposed as supervised validation of generated stories.

The underlying idea is based on the way in which humans learn to write stories. In order to write stories that are acceptable by an audience, a writer must *learn* how to write. Without the intention of developing a model of how this is learnt, it can be intuitively observed that writers must receive feedback from the audience to improve their skills. Without this feedback, human writers are not able to develop their abilities.

Making an analogy, and not trying to mimic the cognitive processes driving human behavior, the model uses human feedback for generated stories just by querying their acceptance about the story. That is, once the story is generated and realised in a human readable form, the human evaluator is asked whether she or he accepts the story as correct.

With this information, the story is classified and the system can adapt itself. The adaptation process is carried out following the next steps:

1. The sequence corresponding to the realised story is analysed and its preconditional links are identified.
2. The preconditional links are abstracted to preconditional rules.
3. The preconditional rules are added to the current rule-set.
4. The current rule-set is used to perform an additional cycle of the process by generation a new story.
5. This list of steps is executed until a particular percentage of stories is rated as correct by the human evaluator.

The addition of the rules to the rule-set is carried out in the way explained in the previous section. Following the previous list of steps, the rule-set keeps being updated through the collection of human feedback.

Preliminary Results

The proposed system has been built as a prototype in which the explained processes have been implemented in a very straightforward way, trying to match, to the possible extent, the definitions previously introduced. The created program was run as a loop in which a simple textual realization of the automatically generated sequences through the use of the rule-set was given to a human user.

The purpose of the test was to test whether the system can adapt itself to generate correct stories, at least for a restricted domain and for a restricted type of stories. Also, it was important to partially check that the required amount of time and effort to adapt to a new domain is significantly lower than the corresponding version based on rule generation by hand. Therefore, the algorithm informally shown in Figure 3 was implemented, and the program was set to stop when the last 5 stories were rated as correct.

The chosen domain consists on short versions of opera plots. The implementation abstracted rules based on the verb of the event, therefore having rules in the form of $love(x?, y?) \wedge love(y?, z?) \rightarrow kill(x?, z?)$. To reduce the required amount of steps to learn significant information, 9 human written operas (adapted and formalized classic operas from Verdi, Bizet, Puccini and Falla) were initially input to the system. Short versions of operas were selected because the amount of verbs is reduced when compared to the amount of available stories. Since themes in Verdi's operas follow similar patterns, operas were chosen for this prototype as a useful resource.

The translation from the opera to a processable formalisation was carried out by hand by the authors. This is a source of influence of authors' knowledge on the final results of the evaluation, and this must be addressed in further research. Formalized short plot outlines of operas are encoded as shown in Figure 4.

ill(violetta)
love(violetta, alfredo)
together(alfredo, violetta)
forces(germont, violetta)
breakup(violetta, alfredo)
despise(alfredo, violetta)
forgive(alfredo, violetta)
die(violetta)

Figure 4: Example formalization of opera (Verdi's La Traviata).

For this implementation, the evaluation was carried out by 14 people, whose ages ranged between 25 and 60. They have no superior studies about narrative, and all of them studied at university. All the resulting plots show approximately the same form, thus only one is shown to exemplify the kind of output. This is assumed to happen because the opera plots are very simple and there is not much room for different opinions in such a simple domain. The effects of a more complex content must be further studied. In such a case, probably a higher number of evaluators will have to be used.

The results show two important facts: first, the incremental process converges to a state in which the proportion of correct stories against non-correct stories is higher than in the first iteration, so it is concluded that the approach is promising, at least for domains with similar characteristics as the short plots of operas.

Second, a *saturation* point has been identified. It has been empirically shown that the amount of useful rules that can be learnt is limited. Approximately after 20 evaluated stories (the number ranges between 16 and 23 stories for the current experiment) the process is able to continue and new stories are generated, but the proportion of correct stories against non-correct stories does not raise. While more in-depth study is required, it is concluded that this is due to domain restrictions (13 types of actions are used) and the way in which story generation is carried out. The application of preconditional rules in generation is too restrictive to be able to generate new content that can give rise to very different rules.

The average required time for reaching the saturation point was approximately 8 minutes and a half. This means an average time of 25 seconds per story, approximately. Although the creation of the rules by hand from the same domain has not been carried out, it is claimed that this pseudo-automatic process is much faster, based on experience. Therefore, the solution is promising according to its objectives. At least, for simplistic domains as the one that has been tested. It is planned to study more complex domains as part of the future work.

Figure 5 shows an example plot representing the proportion between coherent and non-coherent stories for the last 5 generations. This is the result of one of the evaluators, and it is included here because it exemplifies well the behaviour of all the participant evaluators. In this plot it can be seen how the first iterations show varying results and, then, between iterations 15 and 21, there is a period in which the system is not able to improve its ratio according to the gathered rules. Finally, after story 21, the system did generate several coherent stories per non-coherent story. After this point, the saturation zone starts. The system was not able to improve its behaviour in this zone.

The acquired set of preconditional rules after the execution of this experiment is large. It contains 108 elements in the set of "good" rules and 10 elements in the set of "bad" rules. For a short sample of the output of the system, next is a fragment of the set of good rules:

$breakup(x?, y?) \wedge together(y?, x?) \rightarrow back(x?)$
 $escape(x?, y?) \rightarrow back(y?)$
 $back(x?) \wedge ill(x?) \rightarrow die(x?)$
 $ill(x?) \wedge kill(x?, z?) \rightarrow die(x?)$
 $breakup(x?, y?) \wedge love(x?, y?) \rightarrow die(x?)$
 $chase(x?, y?) \wedge love(x?, y?) \rightarrow die(x?)$
 $despise(y?, x?) \wedge forgive(y?, x?) \rightarrow die(x?)$
 $kill(x?, y?) \wedge love(x?, y?) \rightarrow die(x?)$

And this is a fragment of the set of "bad" rules:

$breakup(z?, x?) \wedge love(x?, y?) \rightarrow die(x?)$
 $despise(y?, x?) \wedge love(x?, y?) \rightarrow breakup(y?, x?)$
 $help(z?, y?) \rightarrow breakup(z?, x?)$

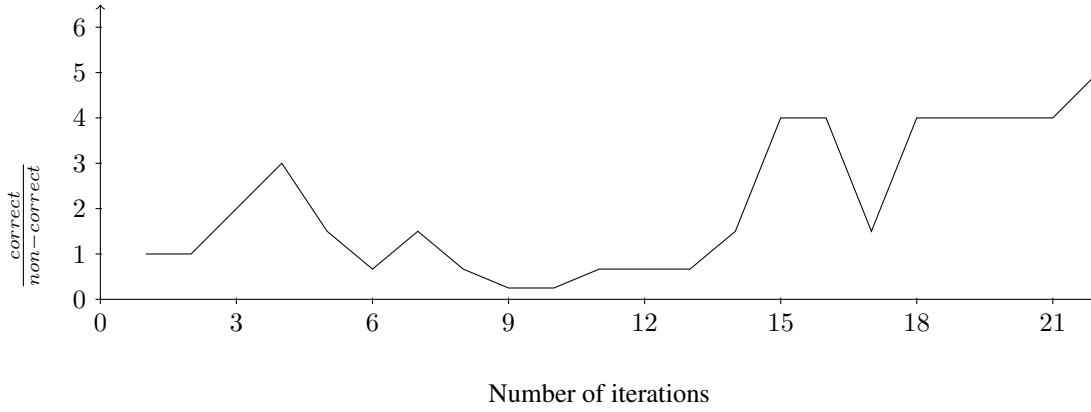


Figure 5: Learning curve from the execution example.

$ill(x?) \wedge kill(z?, y?) \rightarrow forces(z?, x?)$
 $want(z?, x?) \rightarrow kill(z?, y?)$
 $ill(x?) \rightarrow love(y?, x?)$
 $together(y?, x?) \rightarrow want(z?, x?)$

Figure 6 shows the average refinement curve gathered from all the experiments. The average relation between correct and non-correct stories is shown. It can be seen how the proportion between coherent and non-coherent stories almost lineally raises during the execution of the tests, which indicates that, for the chosen working domain and according to the proposed experiments, the system is “learning” a set of preconditional rules that creates coherent stories. This is done without crafting the rules by hand.

Discussion

This proposal does not intend to *replace* semantic processing in computational narrative. Much work has been done so far and all the gathered knowledge is very valuable. Having identified this particular barrier in knowledge acquisition, structural analysis only offers additional tools. In general, the authors hypothesize that the joint effort of these two approaches can be key to successful story processing in the large.

Narratives are based on many more aspects than plain structure and this is accepted by the authors. Human interpretation, viewpoint, emotions and other concepts are heavily involved in our conception of narrative, and these are not representable in term of a lineal structure. Therefore, the proposed system is unable to tackle all aspects of narrations in its current form.

Regarding other work in structural study of narrative (Chambers and Jurafsky 2008), we propose the present study as complementary content. The work by Chambers and Jurafsky seems very promising, but (just as this one does) it has its own intrinsic limitations. In particular, we think that the short script narrations considered by Chambers and Jurafsky lack the properties of full stories, which are addressed in this work by the use of a definition of coherence that takes them into account. Additionally, the outcome of the learning process in our proposed solution are rules, and they can

be edited by hand if needed in a more fine-grained way than complete scripts because the former are slightly more independent from the rest of the learnt content. All these ideas and the relation between these two projects must be further studied.

Also related to that research, it has to be said that the current proposed model in this paper is only able to tackle short, simple stories. Processing of complex narrative patterns are still outside the capabilities of the learning algorithm. Therefore, there is as yet only a small difference between the particular forms that short stories and narrative schemas take. However, something that is considered very important is handled in this work: since some sense of coherence is proposed, stories must be *complete*, they have to be totally coherent as a story, and not only as narrative fragments or scripts.

An important drawback of this approach is the extent to which it can lead to generation of very good stories. From a general point of view, the change of perspective from semantic management of information to structural processing implies some loss of expressive power. Whatever particular corpus is used, there can always be some structures that will never be learnt. These are likely to be those structures that are not commonly used. And, sometimes, these are specially the exceptional ones. Therefore, exceptional quality is hard to reach through structural or statistical methods.

To support the argument of the use of the surface analysis approach, it could be claimed that hand-crafting rules can more easily lead to these seldom present structures. But, particularly in the case of story generation, we claim that story generation has not yet reached the point where exceptional stories are the target, and we hypothesize that this is due to the current perspective about how story generation should be addressed, at least in part.

It has been assumed that the structural definition of preconditional links empirically describes correctness for stories, but many other definitions would be valid, specially taking into account that cognitive models are intentionally avoided. Following this approach, any definition yielding good results in evaluation could be valid. This perspective



Figure 6: Average proportion of coherent vs. non-coherent stories during the refinement process. The proportion of the last 5 stories is shown.

of Artificial Intelligence has produced useful results in some fields like Information Retrieval (Korfhage 1997), and this works hypothesizes that it could be useful for computational narrative to study the possibilities of this kind of models.

It is not claimed, however, that structural or surface form applications *only* can lead to human-like works. But a hybrid approach could make it possible to process stories in large numbers, which has been an long standing goal of the computational narrative community. Models like the one proposed in this paper and others try to unlock the hypothetical potential of non-cognitive solutions.

Additionally, it must be taken into account that creating narrations that are completely human-like need not be the only objective of computational narrative. Completely mimicking human behaviour is arguably useful, but computer generated stories that are recognizable as computer works could also be useful, perhaps in fields or aspects of society beyond those where classic narrations are currently employed. This, of course, should be further discussed in depth.

The issue of saturation is worth mentioning. It intuitively captures the idea of having gathered all possible information from a domain, even if this information does not handle the conceptual meaning of the concepts involved. Indirectly tweaking the threshold of saturation by modifying the way in which stories are automatically generated, for instance, or perhaps introducing some kind of noise into the system in order to slightly prevent local maxima, could heavily affect the behaviour of the system.

If saturation happens too early, not much information would have been collected, but the efficiency of the system could be considered high because only a small amount of time would have been spent. On the other hand, the capabilities of the final rule-set would be quite limited because it is likely that only a small amount of relations was learnt.

If saturation happens too late, the amount of processed stories would have been large, so the quality of the rules could be, in principle at least, high. But that would have required a long training process, thus having consumed significant amounts of user's time, perhaps to the point where other approaches would have been more efficient.

Concluding, it makes sense to study the possibilities of searching for an optimum point of saturation in which the quality of the rules is acceptable but the required time to gather them is correctly adjusted to the user's availability.

It has been previously said that the current approach intends to keep the semantic load of computing narrations to a minimum. In order to realize this, a synthetic relation has been defined with no underlying cognitive model. However, this relation, the *preconditional link*, has been inspired by the way in which humans apply heuristic knowledge when interpreting and understanding stories. For instance, the definition of the preconditional link establishes that for a list of events to be preconditionally linked to another one, they must appear *before* in the story. This suggest that human behaviour has been partially included in the definition, since humans tend to assign causal and chronological links between near events in a story.

Therefore, the extent to which the proposed model only considers surface form of narratives must be further studied and discussed. While it is claimed that surface structure can be valid as a complementary improvement of the efficiency of computational narrative systems, it is clear that the separation between cognitive and structural is not totally defined in formal terms.

Conclusions and Future Work

This paper has presented a proposal for complementing classic approaches to computational story processing, which are based mainly on semantic approaches, with techniques based on structural content of narrations. While the presented model is oriented to story generation, the concepts can be applied to story understanding. The results of a prototype implementation of the theoretical ideas underlying the model is shown, and the main conclusions are presented and discussed.

Much is still to be done in this area. The presented model is only applicable for short stories, which, while containing a closed message, are still too similar to narrative scripts. The preliminary results are promising, but it is clear that the cho-

sen domain does not include any sophisticated narrative features such as humans use. Future work contemplates studying new approaches to broaden the scope and applicability of this kind of solutions.

The implementation has involved certain steps where knowledge has been inserted in the loop, for instance, the translation of the source stories to formal sequences by hand. At this stage, authors' intuition has affected the empirical results, and this must be improved for future versions.

In contrast, the simplistic implementation of the steps of the model has provoked very low saturation levels in the rule gathering execution. More sophisticated implementations could be carried out in order to inform the algorithm. This would mean inserting knowledge in the system, therefore creating a kind of hybrid approach between cognitive and structural approaches. Authors hypothesize that this could be the way to achieving efficient systems for computational narrative.

Additionally, it is important to discuss what are the limits of structural and cognitive approaches in narrative. Applying knowledge from narratology could help to broaden the perspective about what these two concepts really mean, what is the relation between them and how to apply them formally in computational narrative.

Acknowledgements

This research is funded by the Ministerio de Investigación, Ciencia e Innovación (MILES: TIN2009-14659-C03, GALANTE: TIN2006-14433-C02-01), Universidad Complutense de Madrid and Dirección General de Universidades e Investigación de la Comunidad de Madrid (IVER-NAO: CCG08-UCM/TIC-4300) and by the Creation and Consolidation of Research Groups Program by Banco Santander Central Hispano-Universidad Complutense.

References

- Barthes, R., and Duisit, L. 1975. An introduction to the structural analysis of narrative. *New Literary History* 6(2):237–272.
- Bringsjord, S., and Ferrucci, D. 1999. *Artificial Intelligence and Literary Creativity: Inside the mind of Brutus, a Story-Telling Machine*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Chambers, N., and Jurafsky, D. 2008. Unsupervised learning of narrative event chains. In *Proceedings of ACL-08: HLT*, 789–797. Columbus, Ohio: Association for Computational Linguistics.
- Chambers, N., and Jurafsky, D. 2009. Unsupervised learning of narrative schemas and their participants. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, 602–610. Suntec, Singapore: Association for Computational Linguistics.
- Charniak, E., and McDermott, D. 1985. *Introduction to artificial intelligence*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.
- Dehn, N. 1981. Story Generation After Tale-Spin. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 16–18.
- Herman, D. 2000. Narratology as a cognitive science. *Image and Narrative*.
- Kelly, G. 1955. *The Psychology of Personal Constructs*, volume I,II. Norton, New York.
- Klein, D., and Manning, C. D. 2003. Fast exact inference with a factored model for natural language parsing. In *In Advances in Neural Information Processing Systems 15 (NIPS)*, 3–10. MIT Press.
- Korfage, R. R. 1997. *Information Storage and Retrieval*. New York, NY, USA: John Wiley & Sons, Inc.
- León, C., and Gervás, P. 2010. The Role of Evaluation-Driven rejection in the Successful Exploration of a Conceptual Space of Stories. *Minds and Machines* 20(4).
- Meehan, J. 1976. *The Metanovel: Writing Stories by Computer*. Ph.D. Dissertation, Yale University.
- Pérez y Pérez, R. 1999. *MEXICA: A Computer Model of Creativity in Writing*. Ph.D. Dissertation, The University of Sussex.
- Propp, V. 1928. *Morphology of the Folk Tale*. University of Texas Press.
- Riedl, M. 2004. *Narrative Planning: Balancing Plot and Character*. Ph.D. Dissertation, Department of Computer Science, North Carolina State University.
- Turner, S. 1992. *MINSTREL: A Computer Model of Creativity and Storytelling*. Ph.D. Dissertation, University of California at Los Angeles, Los Angeles, CA, USA.