Introducing Mood and Affinity to Generate Brief Template-based Dialogues in Storytelling Systems

Alejandro OÑATE [a,1], Gonzalo MÉNDEZ [a,b] and Pablo GERVÁS [a,b]

[a] *Facultad de Informática, Universidad Complutense de Madrid*

[b] *Instituto de Tecnología del Conocimiento, Universidad Complutense de Madrid*

Abstract. This paper describes the earliest stage of our work on generating dialogues for stories. It aims to develop an algorithm that allows us to create short character-to-character dialogues in Spanish and in controlled contexts. For this dialogue generation, a system based on RiveScript templates has been designed. The main goal is to generate brief question-answer dialogues taking into consideration characters' mood and affinity. These dialogues are generated based on a selected context, together with some input characters' parameters. In this way, it allows us to create brief dynamic conversations that evolve together with the characters.

Keywords. dialogue generation, templates, mood, affinity, story generation

1. Introduction

In the field of Artificial Intelligence (AI), automatic story generation has been the object of research for more than fifty years. The Story Generation Algorithms (SGA) are computational procedures with a textual output that can be considered a story. Currently, there is a large number of storytelling systems in existence [1].

However, these algorithms have a shortcoming: they only create monologues or indirect speech text, even though in some cases these algorithms can build fairly sophisticated stories.

In most interactive storytelling systems, the ability to generate dialogues between characters [2] is still not present, even though a good story can benefit from the inclusion of conversations in the story, which could help readers understand better the characters', allowing them to engage more with the story.

Traum [3] states that, currently, dialogue management systems do not face novel interactive behaviors. The current trend is to base dialog generating systems on the human analysis of data and the updating of models, instead of letting the system develop a new understanding and capacities through the use of dialogue.

The main goal of this work is to develop a system able to generate dialogues that can enrich a story in Spanish. To carry out this task, we have focused on dialogue generation,

[1] Alejandro Oñate: Facultad de Informática, Universidad Complutense de Madrid,
C/ Profesor José García Santesmases 9, 28040 Madrid, Spain; E-mail:aleonate@ucm.es.

with a question-answer style. In these first steps, these dialogues are generated using a template-based mechanism.

In this way we can quickly obtain results and we are able to focus on the additional value of these dialogues, which is in the speech style of each character. In order to generate dialogues, we take into account the mood of each character and the affinity that exists between them. This allows us to generate different dialogues for the same situation, taking into account the current characteristics of each character.

2. Related Work

There are several methods to address the task of dialogue generation. Currently, systems based on deep learning techniques are booming. Neural models of dialogue generation are able to generate dialogues between conversational agents. These techniques are improving very quickly and they already produce realistic dialogues. In the work described in [4] they offer continuity and coherence between past and future dialogues.

These techniques rely on the compilation of large volumes of data of examples of dialogue. This corpus of dialogues, in order to be valid, must be correctly labeled with the knowledge that is to be used as input parameter of the generation system. This is a very laborious task, and resources along these lines are usually only available for the English language, since English is the most commonly used language in this research area.

A quick and effective method to obtain these large volume of dialogues and build the corpus automatically is to use databases of film subtitles [5]. This technique is widely used and helps in the generation of realistic dialogues, but it does not lend itself to be driven by character mood or affinity, since it is difficult to obtain values for these features automatically from the dialogues.

Since our main objective is to enrich dialogues by using characters' mood and affinity, the task of creating the corpus increases enormously, as the corpus should be composed of thousands of dialogues in Spanish, in different situations and different contexts. In addition, in order to be exhaustive, all of these dialogues should be generated and labelled for all the possible combinations of personality and mood of the involved characters.

For this reason, this work focuses on techniques for dialogue generation based on templates, similarly to the work described in [6]. With systems of this type we have a greater control over the generated text, and the aesthetics of the resulting dialogues are usually better than those obtained using other options. This is an important factor, since it allows us to generate a wide range of dialogues using configuration parameters without the need to develop a large corpus.

Cavazza and Charles [2] have carried out some work on dialogue generation in interactive storytelling where they show an interactive storytelling approach which integrates dialogue generation within narrative situations. The first step is focused on invitations, requests for resources (borrowing objects, etc.) or requests for assistance and they generate dialogues where each character takes part in the conversation once. The generation of these question/answer dialogues is based on a template system. In addition, they also focus on the genre and dramatization of a story, highlighting that the expressions of a comedy are different from those of a drama, and the authors mention the use of affinities.

In their examples, only the role of the characters is mentioned as affinity, which can just have a positive or negative value.

The described method is quite different from what we intend to do, since our purpose is to use the affinity value by defining it as a continuous gradual value, rather than a discrete state of just two values (positive or negative).

Another important reference in the field of dialogue generation is the game *Façade*, by Mateas and Stern [7], which can be considered a dialog generation system in video game format. This system relies on the internal representation of the affinity between two characters, Grace and Trip. The authors continued working on this system and extending its capabilities over the following years until July 2005, when the video game in 3D was released. The video game is set in the apartment of the couple, Grace and Trip. While enjoying a perfect evening, both start arguing and a conflict arises which leads to the affinity between them to start shifting. The player's role in the game is to talk to them in order to help them find a solution to their problem. During the game, the players can see how their judgments can influence the status of the characters making their affinity change for better or worse.

It is important to emphasize that Façade makes use of different Natural Language Processing techniques, and among them we can find the use of templates, which was an additional indication that led us to consider this mechanism as a starting point for our work.

As a first step we will carry out a proof of concept where a system based on templates will be used as a generation mechanism. Once we manage to have positive enough results, templates are to be substituted by more sophisticated Natural Language Generation techniques and tools.

Nowadays there is a great variety of systems based on templates. The best known among them is, in all probability, AIML (Artificial Intelligence Mark-up Language) [8], designed to create ALICE, the first Artificial Linguistic Internet Computer Entity Chatterbot. It is a powerful system that allows the creation of a wide variety of dialogue templates and that has been improved in its version 2.0. The first prototypes for the work reported in this paper were created using AIML 2.0, and although the obtained results were acceptable, further research and comparison with alternatives to AIML showed us that there were newer and more powerful options available that included features better suited to our needs.

The system that was finally chosen is RiveScript[2], which covers the same needs as AIML did and adds some additional features that have been decisive for our decision:

1. It allows inheritance between topics, which is something really useful to chain routines.
2. Creation of variables, lists of values, and macro functions in Perl, Python and JavaScript.
3. Weighted random responses and random sub-sentences.
4. It supports numerous comparison operators. While AIML only allows equal, RiveScript allows operations such as equal, not equal, less/greater than and less/greater than or equal.

[2] https://www.rivescript.com/

3. Dialogue Generation using Templates

Nowadays dialogue-based system that use templates are extensively used to implement chatbots that interact with people. Every day, the amount of chatbots in the messaging applications or websites that we use is constantly increasing. These bots have a defined action list they can perform. Such a controlled context allows us to create dialogue templates and to answer to every question the user asks in a natural way.

Unlike in bot-to-person systems, our aim is to generate dialogues to enrich stories without the participation of a user. The two interlocutors are fictional characters that take part in the story, created by us. This is a relevant aspect when creating the templates that are going to be used to generate the dialogue, as it allows us to control what input sentences our character can receive, and these input sentences can subsequently be modelled as output sentences from the previous character. For this reason, both characters share the same knowledge base.

3.1. Template Structure

Elaborated templates are not conventional templates that a chatbot can normally use. These templates sometimes use commands as input sentences that allow us to control the conversation flow. A clear example of this is the beginning of a conversation, which is always attended by the so-called routing template that analyzes the initial command and decides which topic of conversation to set.

Once the topic has been selected, the conversation begins. Each topic uses templates that serve specific questions and answers about that topic. In this way, the templates have been created in a generic way encompassing recurring activities. These templates receive an action (verb) and a complement (noun or direct complement) as environment variables. This type of parameterized template allows us to reuse a common structure within the same topic. This configuration is what we have called "environment variables". They are series of parameters unknown to the characters and that do not affect the development of the conversation. The importance of the mentioned variables is that, in the context of the story that we are enriching, we can find the generated dialogue.

An example would be the "schedule a plan" topic, in which we can create a dialogue where one person suggests a plan to another, independently of the plan, and the second person accepts or rejects the suggestion. To achieve this, it is only necessary to configure the action and the complement.

> *In Spanish:*
> [ir+(cenar, parque) / jugar+(videoconsola, ajedrez) /...]
> *In English:*
> [go+(dine, park) / play+(video game console, chess) /...]

In addition, some dialogues may have other "environment variables" which offer added value to the conversation. Following the "schedule a plan" topic, it is possible that we want to add new information to our context. In case of accepting the invitation, one of the characters proposes a "day" and a "time" to perform the action. In case of a rejection, the character proposing the plans may persistently try to convince the other by proposing another day or another time to carry out the plans.

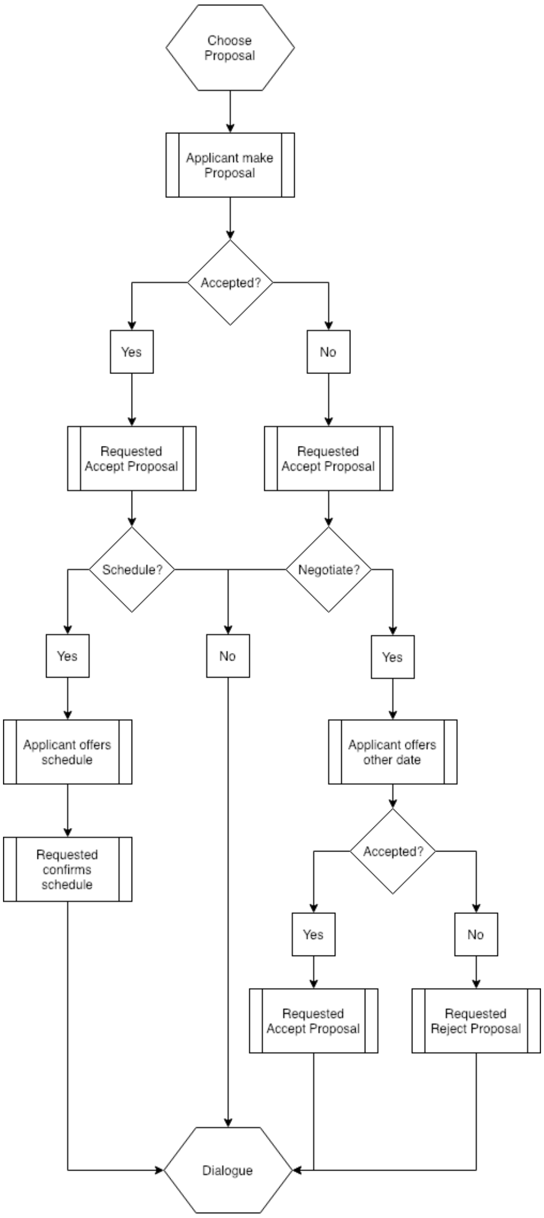
Choose Proposal
Applicant make Proposal
Accepted?
Yes
No
Requested Accept Proposal
Requested Accept Proposal
Schedule?
Negotiate?
Yes
No
Yes
Applicant offers schedule
Applicant offers other date
Requested confirms schedule
Accepted?
Yes
No
Requested Accept Proposal
Requested Reject Proposal
Dialogue

Figure 1. Depiction of the flowchart of a proposal

In these cases, "day" and "time" are environment variables that our system receives as inputs to contextualize with the main story. We can see examples of the flow in Figure 1 and the variables in section 3.4.

3.2. Character Parameterization

The characters of the story are parameterized to lead the conversation in the way we need. The first and simplest parameter determines whether the recipient character will accept or reject the proposal. This determines the resolution of the proposal.

We have also added the option to extend or not the conversation in order to determine extension and additional dialogue content. For example, if a character decides to extend the dialogue after accepting a proposal to "schedule a plan", the dialogue starts a confirmation stage of the day and time of the event, whereas if a dialogue is extended after rejecting a proposal, the characters try to negotiate another day and time to carry out the activity.

On the other hand, there are parameters that can affect the speech style of the characters. Among these parameters we can find the character's mood. Currently, this parameter is a discrete value and allows us to set three possible values: sad, neutral and happy. Depending on the mood of each character, they will select their questions or answers from a different sentence catalog. These sentences have been previously categorized into the three moods. We believe that mood is interesting not just to help us decide whether to accept or reject a proposal, but also to select the kind of expression that a character will use. A happy character will accept a proposal using a sentence like "Yes, I really like it" while a sad character is less likely to accept a proposal and, if she does, she could answer something similar to "Well... that's okay."

Once a character has an assigned mood, the second parameter comes into play. This parameter refers to the affinity between the two characters [9] which take part in the conversation. The affinity value is continuous, so it can take values from 0 to 100. It is important to note that this parameter is not reciprocal, that is, the affinity of Character A with Character B may have a different value than the affinity of Character B with Character A. The aim of this parameter is to define affinity thresholds for sentences of each mood catalog. The characters will choose sentences with the highest value that is below the affinity value that governs the interaction in question (if the affinity is greater than a given threshold). Highly ranked sentences are used for characters with a strong emotional bond, where they will result in sweet and affectionate expressions, while in the lower ranks the characters will speak in a cold, more direct way.

At the end, to provide more realism to the dialogue, within each sentence catalog of mood and affinity threshold we can find more than one sentence, which provides more variability to the generated dialogues. These sentences are weighted with values from 0 to 100 with the objective of sorting the priority of the sentences. In this way, if we reach the same point during the same conversation with the same values of affinity and mood, we will obtain different sentences creating a less repetitive dialogue.

3.3. Post-processing Answers

In order to obtain a more real and credible conversation, a post-processing of the answers is necessary. One of the techniques used is to discard sentences already used. Since the two characters taking part in the conversation share a common knowledge base, and provided that in some negotiation stages they can match in the same response block, a simple control mechanism that allows us to discard a response that has already been used by other characters has been implemented.

In addition, a verbal processing system has been included in order to achieve a greater variety of expressions, using as a configuration parameter the fact that the action to be performed is a verb in infinitive. This allows us to conjugate these verbs, so in the template we use the verb environment variable followed by the conjugation we need. For example:

Template:
$[< env\ verb > |PRESENT|1|PL|SUBJUNCTIVE|IMPERFECTIVE]$
Spanish conjugation:
ir a cenar → vayamos a cenar
English conjugation:
to go to dinner → let us go to dinner

This is an important point since our system is designed to generate sentences in Spanish, which has some very complex verb conjugations.

3.4. Example of Operation

Several dialogues with different moods and affinities have been launched in order to put into practice the template-based genration that we have defined throughout this article. After several tests on the system, affinity thresholds established on template sentences have been empirically adjusted in order to obtain a response according to their value and human language. In this way we have assigned a mood and affinity threshold to each option of the template.

Once the system remains stable with its response, variations in character’ mood and affinity have been incorporated. To carry out this task we were inspired by [5], creating a simple system that makes adjustments in the affinity values according to previous dialogue resolution. This algorithm allows us to make variations in the values of mood and affinity with the intention of simulating how the dialogues of two characters would be if their affinity and mood varied according to their previous experiences.

The affinity between the characters is the criterion of greatest weight when defining whether a proposal is accepted or rejected. Both affinity values have been taken into account, as we saw that the characters can perceive different affinities with each other.

The acceptance or rejection criterion is based on the affinities between the character who is making the proposal (applicant) and the one which is receiving this request (requested), where the contribution in taking the decision is of 70% for the latter (requested) and 30% for the former (applicant).

The affinity update is performed with a logarithmic increment function obtained empirically, in order to make the affinity vary faster when the values are low and slower when the values are high. We made this decision because the more mature affinities are, the more stable the relationship is and the more tolerant people are in case of a rejection. On the other hand, when characters are cultivating a relationship or when they always receive rejections, the positive factors generate greater satisfaction.

The new affinity is obtained by increasing or decreasing the current affinity, according to whether the proposal was accepted or rejected with a value that is obtained in the following way: the inverse of the natural logarithm of the current affinity multiplied by

Table 1. Example of Affinity Variation

Applicant	Requested	Resolved	Applicant new	Requested new
Alice 68	Bob 54	accept	Alice 79	Bob 59
Alice 79	Bob 59	reject	Alice 68	Bob 54
Bob 54	Alice 68	accept	Bob 66	Alice 72
Alice 72	Bob 66	accept	Alice 83	Bob 70
Alice 83	Bob 70	accept	Alice 93	Bob 74
Alice 93	Bob 74	reject	Alice 83	Bob 70
Bob 70	Alice 83	accept	Bob 81	Alice 87
Alice 87	Bob 81	accept	Alice 97	Bob 85

a factor “C”, with the value of “C” empirically set to 60 in the case of the applicant and 40 in the case of the requested

$$\boxed{nA = A + \frac{1}{\ln(A)} * C}$$

In addition, this variation affects indirectly the character who receives the proposal. We can see an example of the evolution of the affinities of two characters during a conversation generated with our system in Table 1.

Once the proposal resolution has been decided, the environment values are set with resolution, moods and affinities. This generates a dialogue about an activity in which the characters talk to each other based on the personality parameterization.

When the dialogue has been obtained, and taking into account the proposal result, the affinity and mood values are readjusted. The affinity increases or decreases taking into account the last resolution while the mood observes the last N resolutions and varies if it has received several rejects or acceptances.

The mood update is carried out when an applicant receives the same response 3 times in a row or a total of 5 responses of the same kind since the last state change. After each change, the counters returns to zero. In Table 2 we can see a simulation of how the counters increase and the moment in which the mood is updated. This simulation is extracted from a conversation created with our system.

In Table 3 we can see a dialogue sequence according to the value of the affinity. In order to show more varied examples, these dialogues are not consecutive in the flow of affinity and mood update.

4. Conclusion

In this paper, the use of templates for dialogue generation in Spanish in the context of story generation systems has been described, along with the use of the characters’ mood and mutual affinity to select the responses in a proposal-like dialogue.

The implemented system has allowed us to create generic templates classified by topics that refer to conversations types, using environment variables that allow us to create dynamic dialogues without the need of a large knowledge base.

Table 2. Example of Mood Variation

Current	Resolved	Successive	Total	New
neutral	accepted	1	1	neutral
neutral	accepted	2	2	neutral
neutral	accepted	3	3	happy
happy	accepted	1	1	happy
happy	rejected	1	1	happy
happy	rejected	2	2	happy
happy	accepted	1	2	happy
happy	rejected	1	3	happy
happy	accepted	1	3	happy
happy	rejected	1	4	happy
happy	rejected	2	5	neutral

These templates have been designed to offer a solution based on characters' mood and affinity that allows the creation of different conversations based on specific parameters of each character. In addition, these templates, together with a personality update mechanism, which can vary the mood and affinity between characters, has allowed us to create adaptive conversations where the reader can perceive the characters' evolution and their relationship throughout the dialogue.

Finally, the dialogue generation algorithm makes use of post-processing techniques that produce a better experience and the generation of a more realistic dialogue by applying a layer of pragmatic restrictions based on context to avoid repetitiveness.

5. Future Work

As the next steps to be carried out, it is expected to improve the dialogue generation algorithm by incorporating new configuration parameters. Among these parameters we have thought of the kind of relationship between characters (friends, lovers, partners, enemies) and the degree of affinity between the characters.

This algorithm will be used to generate dialogues throughout a story, adapting them to the context and the personality and affinity parameters.

The last step is to improve the dialog generation algorithm by applying free text generation mechanisms in order to build a Hybrid Natural Language Generation for Spoken Dialogue Systems[10]. The generation algorithm follows a hybrid approach, combining finite state machines (FSM), grammars and the system based on templates.

6. Acknowledgements

The work presented in this paper has been partially funded by the projects IDiLyCo: Digital Inclusion, Language and Communication, Grant. No. TIN2015-66655-R (MINECO/FEDER) and InVITAR-IA: Infraestructuras para la Visibilización, Integración y Transferencia de Aplicaciones y Resultados de Inteligéncia Artificial, UCM Grant. No. FEI-EU-17-23.

Table 3. Example of Generated Dialogues

Original Dialogue in Spanish	Dialogue Translated into English
Bob [neutral,66]: Alice puedes ver una película conmigo?	Alice can you watch a movie with me?
Alice [happy,72]: Será un placer ver contigo una pelicula Bob	It will be a pleasure to watch a movie with you Bob
Alice [sad,54]: Bob quieres visitar a Charlie conmigo?	Bob, do you want to visit Charlie with me?
Bob [neutral,69]: Alice, no tengo muchas ganas de visitar a Charlie.	Alice, I do not really want to visit Charlie
Bob [happy,68]: Te gustaría que visitáramos la ciudad Alice?.	Would you like us to visit the city Alice?.
Alice [neutral,54]: No, gracias Bob. Mejor otro día.	No, thanks Bob. Maybe another day.
Bob [happy,46]: Que día te apetece visitar la ciudad?	What day do you feel like visiting the city?
Alice [neutral,62]: Creo que el viernes por la tarde puede ser un buen día.	I think Friday afternoon can be a good day.
Bob [happy,68]: Genial Alice! me parece perfecto.	Great Alice! it seems perfect.
Bob [happy,79]: Te apetece jugar al ajedrez conmigo Alice?	Do you want to play chess with me Alice?
Alice [sad,85]: Oh... no me apetece jugar al ajedrez. Lo siento mucho Bob	Oh... I do not feel like playing chess. I'm so sorry Bob
Bob [happy,68]: Quieres que juguemos mañana mejor?	Do you want us to play tomorrow better?
Alice [sad,81]: Tal vez Bob, mañana lo vemos.	Maybe Bob, tomorrow we see it.
Alice [neutral,90]: Bob, podrías comprar el periodico?	Bob, could you buy the newspaper?
Bob [happy,54]: Claro Alice, yo compraré el periodico.	Sure Alice, I'll buy the newspaper.
Alice [happy,54]: Puedes ir a la tintorería hoy Bob?,	Can you go to the cleaners today Bob?
Bob [happy,90]: Vaya! No creo que hoy pueda Alice.	Oh! I do not think I can do it today Alice.
Alice [happy,46]: No pasa nada Bob. Que día crees que podrías ir?	It's okay Bob. What day do you think you could go?
Bob [happy,80]: Creo que mañana me sería posible ir.	I think tomorrow it would be possible for me to go.
Alice [sad,71]: Bob te apetecería que fuesemos a tomar un helado mañana?	Bob, would you like us to go for an ice cream tomorrow?
Bob [happy,67]: Por supuesto que sí Alice. A que hora?	Yeah! of course, Alice. What time?
Alice [sad,83]: A las 5:00 pm.	At 5:00 p.m
Bob [happy,73]: Perfecto! allí nos vemos	Perfect! see you there!

References

[1] Pablo Gervás. Story generator algorithms. *The Living Handbook of Narratology*, 19, 2012.

[2] Marc Cavazza and Fred Charles. Dialogue Generation in Character-based Interactive Storytelling. In *Artificial Intelligence and Interactive Digital Entertainment*, pages 21–26, 2005.

[3] David Traum. Computational approaches to dialogue. *The Routledge Handbook of Language and Dialogue*, page 143, 2017.

[4] Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. Deep Reinforcement Learning for Dialogue Generation, 2016.

[5] Marilyn A Walker, Grace I Lin, and Jennifer Sawyer. An Annotated Corpus of Film Dialogue for Learning and Characterizing Character Style. In *Language Resources and Evaluation Conference*, pages 1373–1378, 2012.

[6] Floris Bex and Chris Reed. Dialogue Templates for Automatic Argument Processing. In *COMMA (Computational Models of Argument)*, pages 366–377, 2012.

[7] Michael Mateas and Andrew Stern. Façade: An experiment in building a fully-realized interactive drama. In *Game developers conference*, volume 2, pages 4–8, 2003.

[8] Richard Wallace. The elements of AIML style. ALICE AI Foundation, 2004.

[9] Gonzalo Méndez, Pablo Gervás, and Carlos León. *On the Use of Character Affinities for Story Plot Generation*, volume 416 of *Advances in Intelligent Systems and Computing*, chapter 15, pages 211–225. Springer, February 2016.

[10] Michel Galley, Eric Fosler-Lussier, and Alexandros Potamianos. Hybrid natural language generation for spoken dialogue systems. In *Seventh European Conference on Speech Communication and Technology*, 2001.