POS-tag based poetry generation with WordNet

Manex Agirrezabal, Bertol Arrieta, Aitzol Astigarraga
University of the Basque Country (UPV/EHU)
IXA NLP Group
Dept. of Computer Science
20018 Donostia
sgpagzam@ehu.es
bertol@ehu.es
aitzol.astigarraga@ehu.es

Mans Hulden
University of Helsinki
Department of Modern Languages
Helsinki, Finland
mhulden@email.arizona.edu

Abstract

In this paper we present the preliminary work of a Basque poetry generation system. Basically, we have extracted the POS-tag sequences from some verse corpora and calculated the probability of each sequence. For the generation process we have defined 3 different experiments: Based on a strophe from the corpora, we (a) replace each word with other according to its POS-tag and suffixes, (b) replace each noun and adjective with another equally inflected word and (c) replace only nouns with semantically related ones (inflected). Finally we evaluate those strategies using a Turing Test-like evaluation.

1 Introduction

Poetry generation is one of the dream tasks of Natural Language Processing (NLP). In this text we point out an approach to generate Basque strophes automatically using some corpora, morphological information and a lexical database. The presented method is not tied to a specific language, but it is especially suitable for inflected languages, as the POS information used in some tasks with success in non inflected languages is not enough for inflected ones. We have used the POS-tags with their inflectional information to learn usual structures in Basque poetry.

This work is part of a more general and complete project, called *BertsoBOT* (Astigarraga et al., 2013). BertsoBOT is a robot capable of creating and singing Basque verses automatically. The robot joins together in a single system techniques from robotics, NLP and speech synthesis and recognition. The work presented in this paper comes to improve the generation module of the mentioned system.

Although our intention is to create whole verses, in this paper we present the first steps towards it: the creation of strophes. Additionally, Basque verses have to rhyme, but in these first experiments we have not considered it.

Basque language

Basque language is spoken along the Basque Country[1] by approximately 700.000 people. Although there is a standardized form of the language, it is common the use of non-standard dialects in certain regions, mainly in spoken language.

Basque is a morphologically rich language, which is an obvious feature if we analyze the multiple declension cases[2] that can be used with only one word. For example, the phrase "with the friends" can be expressed with only one word, "*lagunekin*".

lagunekin = *lagun* (friend) + *ak* (plural determiner) + *kin* (with)

Art of *bertsolaritza*

The art of *impromptu* verse-making, *bertsolaritza*, is very ingrained in the Basque Country. The performances of verse-makers are quite usual and a big championship is held every four years which congregates 15.000 people, approximately. One tipical work to do for the verse-makers is to sing verses extempore, given a topic. The particularity of these verses is that they have to follow strict constraints of meter and rhyme. In the case of a metric structure of verses known as "*zortziko txikia*"

[1] http://en.wikipedia.org/wiki/Basque_Country_(greater_region)

[2] en.wikipedia.org/wiki/Basque_grammar#Declension

Proceedings of the 14th European Workshop on Natural Language Generation, pages 162–166,
Sofia, Bulgaria, August 8-9 2013. ©2013 Association for Computational Linguistics

(small of eight), the poem must have eight lines. The union of each odd line with the next even line, form a strophe. Each strophe, has a small structure[3] and must rhyme with the others. Below, you can see an example of a verse, with *lauko txikia*[4] stanza:

Neurriz eta errimaz	With meter and rhyme
kantatzea hitza,	to sing the word
horra hor ze kirol mota	bertsolaritza is
den bertsolaritza.	that kind of sport

2 State of the art

A good review of computer guided poetry can be found in (Gervás, 2010). Most relevant ones include:

WASP

The WASP system (Gervás, 2000) can be considered one of first serious attempts to build an automatic poetry generator system. It is based on the generate-and-test paradigm of problem solving. Simple solutions are generated and then coupled with an evaluation function for metric constraints, producing acceptable results.

ASPERA

ASPERA (Gervás, 2001) is a case-based reasoning (CBR) system for poetry generation. It generates poetry based on the information provided by the user: a prose description of the intended message, a specific stanza for the final poem, a set of verse examples on that stanza, and a group of words that the final poem must contain.

The system was implemented using CLIPS rule-based system, and follows the four typical CBR steps: Retrieval, Reuse, Revise and Retain.

POEVOLVE

Levy (Levy, 2001) went on to develop an evolutionary model of poetry generation. POEVOLVE creates limericks taking as a reference the human way of poetry writing. The POEVOLVE system works as follows: an initial population is created from a group of words that include phonetic and stress information. Rhymes that meet the requirements are selected and then more words are selected to fill the rest of the verse-line based on their stress information. A genetic algorithm is employed to modify the words that compose the limerick. Evaluation is performed by a neural network trained on human judgements. It must be said that this system does not take syntax and semantics into account.

McGonnagall

Manurung presented also an evolutionary approach to generate poetry (Manurung, 2003). The poem generation process is formulated as a state space search problem using stochastic hill-climbing. The overall process is divided in two steps: evaluation and evolution. During the evaluation phase, a group of individuals is formed based on initial information, target semantics and target phonetics. This group of initial individuals is then evaluated taking into account different aspects such as phonetics, semantics and surface form. Each individual receives a score, and in the evolution step, the subset with higher scores is selected for reproduction. The resulting mutated individuals derive, hopefully, in better versions of the poem.

3 Creating strophes

Our goal is to create Basque strophes automatically. But strophes written by combining words randomly usually do not have any sense. For words have any meaning when combined together, they must be organized following particular patterns. Towards this end we have applied and tested different methodologies. We use a morphological analyzer to extract POS and inflection patterns in strophes, and to create new ones following those schemes. The idea is to find the most commonly used patterns so that we can use them in new strophes. We also improve the results taking semantics into account. In the next lines we are going to describe some resources we have used.

3.1 Corpora

For the learning process of the usual POS-tag patterns we have employed some Basque verse corpora yielded by the Association of the Friends of Bertsolaritza[5] (*AFB*). Those are impromptu verses sung by Basque verse-makers and the transcriptions of this collection have been done by members of the information center[6] of the *AFB*.

For this work, we are going to exploit three corpora,

[3] 13 syllables with a *caesura* after the 7th syllable

[4] Lauko txikia: The same as *zortziko txikia* but with four lines, instead of eight.

[5] http://www.bertsozale.com/en

[6] http://bdb.bertsozale.com/en/orriak/get/7-xenpelar-dokumentazio-zentroa

each one following a classic stanza in Basque verses: (a) small stanza, (b) big stanza and (c) habanera.

a) Small stanza

This corpus has approximately 10.000 lines. Each line of this corpus is composed by a strophe containing 13 syllables with a *caesura* between the 7th and the 8th syllable. This stanza is used to sing sprightly verses composed by compact ideas.

b) Big stanza

In this case, this corpus has about 8.000 lines and each line has 18 syllables with a caesura after the 10th syllable. Depending on the chosen melody, this stanza can also have a complementary pause in the 5th syllable. The topics of this type of verses tend to be more epic or dramatic.

c) Habanera

This corpus has just about 1000 lines and they are composed by 16-syllable lines with a caesura after the 8th syllable. It is commonly used when the verse-maker has to compose a verse alone about a topic.

3.2 POS sequence extraction

To extract the POS-tags, we use a Basque analyzer developed by members of IXA NLP group (Aduriz et al., 2004), which involve phrasal morphologic analysis and disambiguation, among other matters.

Once calculated the POS-tags, we estimated the most probable POS sequences using POS-tag ngrams. We did this in order to know which POS-tag sequence would better fit for each stanza. For example, an acceptable POS-tag sequence in the small stanza corpus would be "NN-NN-JJ-VB". This pattern could be extracted from this strophe, which is correct.

Mirenekin+NN *zakurra*+NN *zoriontsua*+JJ *da*+VB.

(With Miren)+NN (the dog)+NN is+VB happy+JJ.

But to have the POS-tag pattern is not enough for a good generation.

Special issues in the categorization of words in Basque

The gist is that Basque is an agglutinative language, so there is plenty information included in the suffixes of the words. Because of that, if we don't retain any information about suffixes, we would lose some important data. In Basque, we can apply declension to nouns, pronouns, adjectives and determiners. Therefore, we need to save the declension case information to do a correct generation. When a set of words compound a noun phrase, only one of the words will be inflected.

Some verbs, when they are part of a subourdinate clause, can also be inflected. In these cases, we have to extract the suffixes of the verb of that clause, because it expresses the type of clause.

All this information is essential if we do not want to lose the meaning of the clause. Below, you can see an example of generation of strophes in Basque using only POS-tags:

Mirenekin+NN *lagunekin*+NN *zoriontsua*+JJ *da*+VB.

(With Miren)+NN (with the friends)+NN is+VB happy+JJ.

As you can see, the phrase "with Miren with the friends is happy" is not grammatically correct. Storing the declension information, that creation would not be allowed and one of the clauses created by the system could be:

Mirenekin+NN_COM *mahaia*+NN_ABS *zoriontsua*+JJ_ABS *da*+VB.

(With Miren)+NN_COM (the desk)+NN_ABS is+VB happy+JJ_ABS.

The addition of the declension information will avoid some grammatical errors in the generation process. But when the changed element is a verb, the system can insert one that does not follow the same subcategorization[7], which will lead us to a grammatical error too. So, changing the verb without more information can be uncertain.

3.3 Semantic information

On the other hand, if we take a look at the last example, it is not correct to say that the desk is happy. To avoid these cases, we posed the use of the Basque WordNet (Fellbaum, 2010) (Pociello et al., 2011). We used it to change words with related ones.

3.4 Morphological generation

Finally, it is important the fact that Basque is an inflected language. So, we need to have a morphological generator (Alegria et al., 2010) to create the corresponding inflected forms of the words. This generator is based on the Basque morphology description (Alegria et al., 1996).

4 Experiments

In this work, we have performed a set of experiments to analyze different strategies for the generation of stro-

[7] The subcategorization indicates the syntactic arguments required or allowed in some lexical items (usually verbs).

phes in Basque. In the following lines, we explain the ameliorations we get in each experiment.

The first experiment creates strophes by inserting words that are consistent with each POS-tag and its inflection information. We first get some of the most common POS-tag sequences and for each POS-tag sequence the application returns two strophes. The first strophe uses words from the same verse corpus to make substitutions. The second one uses words from the EPEC corpus (Aduriz et al., 2006).

The second experiment creates clauses, but changing only the nouns and adjectives from original strophes from the corpus. We mantain the inflection information. In this experiment we also get two strophes for each pattern sequence, as in the previous attempt (verse corpus and EPEC corpus). With this constraint we avoid the creation of incorrect strophes because of the problem of subcategorization (explained in section 3.2).

The third experiment makes small changes in the original strophes (from the corpus), as it only replaces each noun for a semantically related noun. The related noun can be: (a) Antonym of the original word or (b) hyponym of the hypernyms of the original word. In order of preference, first we try to change each name with one of its antonyms. If there is no antonym, then we try to get the hypernyms of the word to return their hyponims. Once the new word has been found, we add the needed suffixes (the same ones that had the words from the corpus) in order to fit correctly in the strophe, using the morphological generator. The change of words with related ones gives us the chance to express semantically similar sentences using different words.

5 Evaluation

Once the experiments were finished, we made an evaluation in order to analyze the quality of the automatically generated strophes. The evaluation of computer generated poetry is nowadays fuzzy, so we defined a Turing Test-like evaluation. We contacted two linguists that had not done any work on this project, so that the evaluation be as objective as possible. We prepared 135 strophes interleaving some created by the machine with others from the corpus. We asked the evaluators to guess if the strophe was done by the machine or by a human. We only draw conclusions using machine-generated strophes, as we want to know how many of them percolate as human-generated ones. In the next table you can see the rate of sentences created by the machine and suposed to be done by humans:

	EXPERIMENT		
Evaluator 1	1	2	3
Percolated as human	0.033	0.259	0.75
Evaluator 2			
Percolated as human	0.333	0.481	0.75

As you can see, according to *Evaluator 1*, the first experiment was not very worthy, as the only 3.3% of the machine generated strophes percolated as human generated ones. The second experiment got better results, and the 26% of the strophes were thought to be human generated ones. As expected, the strophes of the third experiment are the most trustworthy ones. The results given by the second evaluator are higher, but the important fact is the increase of the progression over the experiments.

6 Discussion & Future Work

In this paper we have presented a set of experiments for the automatic generation of poetry using POS and inflectional tag patterns and some semantics. In the last section we show the Turing Test-like evaluation to measure the reliability of each experiment. This will be part of a whole poetry analysis and generation system.

In the future, we intend to change verbs from strophes controlling the subcategorization of them in order to enable the creation of well-formed strophes about a constrained topic. Also, we plan to use a frame semantics resource, such as FrameNet, and after creating a strophe, make some modifications to get an acceptable semantic meaning.

References

Aduriz, I., Aranzabe, M., Arriola, J., de Ilarraza, A., Gojenola, K., Oronoz, M., and Uria, L. (2004). A cascaded syntactic analyser for Basque. *Computational Linguistics and Intelligent Text Processing*, pages 124–134.

Aduriz, I., Aranzabe, M. J., Arriola, J. M., Atutxa, A., de Ilarraza, D. A., Ezeiza, N., Gojenola, K., Oronoz, M., Soroa, A., and Urizar, R. (2006). Methodology and steps towards the construction of EPEC, a corpus of written Basque tagged at morphological and syntactic levels for automatic processing. *Language and Computers*, 56(1):1–15.

Alegria, I., Artola, X., Sarasola, K., and Urkia, M. (1996). Automatic morphological analysis of Basque. *Literary and Linguistic Computing*, 11(4):193–203.

Alegria, I., Etxeberria, I., Hulden, M., and Maritxalar, M. (2010). Porting Basque morphological grammars to foma, an open-source tool. *Finite-State Methods and Natural Language Processing*, pages 105–113.

Astigarraga, A., Agirrezabal, M., Lazkano, E., Jauregi, E., and Sierra, B. (2013). Bertsobot: the first minstrel robot. *6th International Conference on Human System Interaction, Gdansk.*

Fellbaum, C. (2010). *WordNet*. Springer.

Gervás, P. (2000). Wasp: Evaluation of different strategies for the automatic generation of Spanish verse. In *Proceedings of the AISB-00 Symposium on Creative & Cultural Aspects of AI*, pages 93–100.

Gervás, P. (2001). An expert system for the composition of formal spanish poetry. *Knowledge-Based Systems*, 14(3):181–188.

Gervás, P. (2010). Engineering linguistic creativity: Bird flight and jet planes. In *Proceedings of the NAACL HLT 2010 Second Workshop on Computational Approaches to Linguistic Creativity*, pages 23–30. Association for Computational Linguistics.

Levy, R. P. (2001). A computational model of poetic creativity with neural network as measure of adaptive fitness. In *Proccedings of the ICCBR-01 Workshop on Creative Systems*. Citeseer.

Manurung, R. (2003). *An evolutionary algorithm approach to poetry generation*. PhD thesis, School of informatics, University of Edinburgh.

Pociello, E., Agirre, E., and Aldezabal, I. (2011). Methodology and construction of the Basque Wordnet. *Language resources and evaluation*, 45(2):121–142.