Máster Universitario en

Ingeniería **C**omputacional y **S**istemas **I**nteligentes

Universidad del País Vasco Euskal Herriko Unibertsitatea

Konputazio Zientziak eta Adimen Artifiziala Saila –

Departamento de Ciencias de la Computación e Inteligencia Artificial

K I S A

I C S

MSc Thesis

"Statistical natural language generation for dialogue systems based on hierarchical models"

Kristina Stemikovskaya

Tutora

MARIA INES TORRES BARAÑANO

Grupo de investigación:

PR&Speech Technology Dep. Electricidad y Electrónica Fac. Ciencia y Tecnología Universidad del País Vasco

informatika fakultatea facultad de informática

KZAA

Septiembre 2014

CONTENIDO

1. INTRODUCTION

Due to the increasing presence of natural-language interfaces in our life, natural language processing (NLP) is currently gaining more popularity every year. However, until recently, the main part of the research activity in this area was aimed to Natural Language Understanding (NLU), which is responsible for extracting meanings from natural language input. This is explained by a wider number of practical applications of NLU such as machine translation, etc., whereas Natural Language Generation is mainly used for providing output interfaces, which was considered more as a user interface problem rather than a functionality issue.

Generally speaking, natural language generation (NLG) is the process of generating text from a semantic representation, which can be expressed in many different forms. The common application of NLG takes part in so called Spoken Dialogue System (SDS), where user interacts directly by voice with a computer-based system to receive information or perform a certain type of actions as, for example, buying a plane ticket or booking a table in a restaurant. Dialogue systems represent one of the most interesting applications within the field of speech technologies. Usually the NLG part in this kind of systems was provided by templates, only filling canned gaps with requested information. But nowadays, since SDS are increasing its complexity, more advanced and user-friendly interfaces should be provided, thereby creating a need for a more refined and adaptive approach.

One of the solutions to be considered are the NLG models based on statistical frameworks, where the system's response to user is generated in real-time, adjusting their response to the user performance, instead of just choosing a pertinent template. Due to the corpus-based approach, these systems are easy to adapt to the different tasks in a range of informational domain.

The aim of this work is to present a statistical approach to the problem of utterance generation, which uses cooperation between two different language models (LM) in order to enhance the efficiency of NLG module. In the higher level, a class-based language model is used to build the syntactic structure of the sentence. In

the second layer, a specific language model acts inside each class, dealing with the words.

In the dialogue system described in this work, a user asks for an information regarding to a bus schedule, route schemes, fares and special information. Therefore in each dialogue the user has a specific dialogue goal, which needs to be met by the system. This could be used as one of the methods to measure the system performance, as well as the appropriate utterance generation and average dialogue length, which is important when speaking about an interactive information system.

The work is organized as follows. In Section 2 the basic approaches to the NLG task are described, and their advantages and disadvantages are considered. Section 3 presents the objective of this work. In Section 4 the basic model and its novelty is explained. In Section 5 the details of the task features and the corpora employed are presented. Section 6 contains the experiments results and its explanation, as well as the evaluation of the obtained results. The Section 7 resumes the conclusions and the future investigation proposals.

2. STATE-OF-THE-ART REVIEW

There are different definitions for NLG. The one that better suits to the aims of the present work is the following: Natural language generation is the process of constructing deliberately some kind of natural language output (speech or text) from a non-linguistic representation in order to meet some specified communicative goals [1].

The first step in building a task-oriented dialog system is collecting domain-specific textual data (*corpus*). There exists a big amount of open access corpus, but only a small part is in Spanish. The challenge is even bigger if a specific domain corpus is needed, like in this specific work, where the research is developed for the transportation domain.

2.1 Corpus annotation

The corpus annotation can be defined as a process of corpus enhancement with various types of additional linguistic information [2]. The main goal of annotation is to make a certain corpus more useful for its subsequent processing. There are different levels of linguistic annotation. The most basic type is part-of-speech annotation (PoS tagging).

Part-of-speech tagging represents a text mark, where each word in the corpus is preceded or followed by an abbreviation giving the word's part of speech and sometimes, some morphological information. A wide range of PoS taggers software is developed by different approaches, being the most common Hidden Markov Models and Dynamic Programming methods. Most of the PoS taggers have been created for English, but also various specific tools exist for Spanish. Some of them are *TreeTagger* [3] which uses the decision tree model, and *SVMTool* [4], based on Support Vector Machines.

A typical simple tag set for the PoS tagging can look like following:

Tag	Meaning	Examples
ADJ	adjective	*new, good, high, special, big, local*
ADV	adverb	*really, already, still, early, now*
CNJ	conjunction	*and, or, but, if, while, although*

Tag	Meaning	Examples
DET	determiner	*the, a, some, most, every, no*
MOD	modal verb	*will, can, would, may, must, should*
N	noun	*year, home, costs, time, education*
PRO	pronoun	*he, their, her, its, my, I, us*
P	preposition	*on, of, at, with, by, into, under*
TO	the word *to*	*to*
V	verb	*is, has, get, do, make, see, run*
VD	past tense	*said, took, told, made, asked*
VG	present participle	*making, going, playing, working*

Table 1 A tag set for the Part-of-Speech tagging

An example of a phrase tagged in terms of this table could be as follows:

```
The/DET quick/ADJ brown/ADJ fox/N jumped/VBD over/P the/DET lazy/ADJ
dog/N
```

Lemmatization is a reduction of the words in a corpus to their respective lexemes – the head word forms. It allows examining frequency and distribution information without looking for all possible forms. An example from a multilingual lemmatized corpus Multext-East, consisting of Orwell's Nineteen Eighty-Four parallel translation in six languages tagged for part-of-speech and aligned to English [5], can look like following, where the string «Ncns» means a common neuter, singular noun

<w lemma="there" ana="Pt3">There</w>
<w lemma="be" ana="Vmis3s">was</w>
<w lemma="another" ana="Dg--s">another</w>
<w lemma="crash" ana="Ncns">crash</w>

Table 2 A sample of lemmatized corpus from a Multext-East project [5].

Phonetic annotation is usually applied for a spoken language corpus, using a phonetic transcription and serves for recorded speech analysis.

Parsing allows bringing high-level syntactic relationship between morphosyntactic categories. It is the most common form of corpus annotation after PoS tagging. Parsed corpora are known as *Treebanks.* The majority of parsing schemes are

based on context-free phrase structural grammar. Parsing could be done by a combination of human and machine annotation.

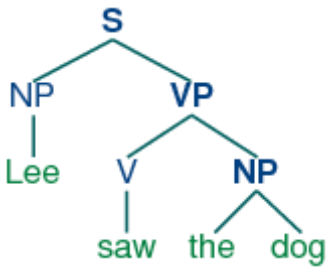

Figure 1 An example of a tree parsing

Semantic annotation could be divided into marking of semantic relationships between items in a corpus and marking of semantic features of words in a text, essentially the annotation of word senses. The best known one is *WordNet* [6], a piece of software which provides access to a semantic network of English.

There are a wide range of instruments that allow the automatic semantic tagging of a corpus.

A powerful tool, which was primarily designed for Spanish language, is *Freeling*, an open-source library which provides lemmatizing, PoS tagging, dependency parsing, etc. for NLP application [7]. Semantic annotation for Spanish in Freeling is based on WordNet usage, in Figure 2 a possible output using Freeling tagger is presented, where “02121620-n” makes reference to a set of synonym dictionary (synset) of WordNet.

El	gato	come	pescado	y	bebe	agua	.
el	*gato*	*comer*	*pescado*	*y*	*beber*	*agua*	*.*
DA0MS0	NCMS000 02121620-n 02122725-n 10022759-n	VMIP3S0 01166351-v 01168468-v 01185304-v	NCMS000 02512938-n 07775375-n	CC	VMIP3S0 01170052-v 01171183-v 01172275-v 01175467-v	NCCS000 04562658-n 07935504-n 09225146-n 14845743-n 14847357-n 15008607-n	Fp

Figure 2 An example of WorNet semantic tagging by Freeling 3.1

Another tool which can be used for semantic tagging is *Phoenix* parser. It is a part of Olympus, a complete open-source framework for implementing spoken dialog systems [8]. The Phoenix parser maps input word strings into a sequence of one or more semantic frames (set of slots, where the slots represent related pieces of information). The developer must define a set of frames and provide grammar rules. Each slot has associated a context-free grammar (CFG) that specifies word string patterns that match the slot.

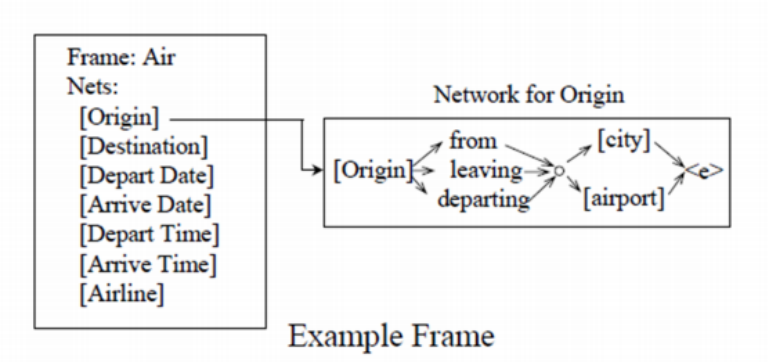

Figure 3 Phoenix frame example []

2.2 Open access corpora in Spanish

As it was mentioned before, there are various corpora in Spanish, freely accessible for research purposes.

One open access corpus is the AnCora-ESP corpus which includes 500,000 words, among them semantically annotated are ~200,000 words [9]. The semantic annotation was realized using *3LB-SAT* [10] annotation tool for WordNet senses tagging, using a sense repository of EuroWordNets-1.6.

Another corpus is the SenSem Spanish [11], which includes texts from the media domain. It includes about a million words (30,000 sentences). The sentences are manually annotated in a syntactic-semantic sense (semantic roles, syntactic functions, syntagmatic categories, constructions, modality and polarity). The query interface is complex and allows searches by verb and / or linguistic phenomenon. It also displays the annotation of searched phrases.

The reason why presented corpora were not used in the present work is the annotation type needed for the construction of semantic classes (described in

Chapter 5). To build the classes the domain-specific manual annotation is required which is a highly time-consuming labor. Thus, only the two previously annotated corpora were used, DIHANA and Bilbobus (detailed description follows in Chapter 5).

2.3 Natural language generation

For modern data-oriented methods, the task of natural language generation can be divided into two steps. First, the so called deep generation is responsible for “what to say” and corresponds to the document planning stage. The second can be summed up as a surface generation, responsible for “how to say” task.

There are multiple approaches to the surface generation problem.

2.3.1 Template-based approach

Template-based generation is the most basic approach and is typically used in industrial dialogue systems, and even in most of the state-of-the-art research systems (e.g. [12]). The Dialogue System response is generated based on templates and canned expressions, which are handcrafted by the developer. The main disadvantage of this approach is that, for any real-life system, the amount of templates that need to be created is counted by thousands. And the system becomes strictly domain-dependent, excluding any possibility for changing the application area [13]. Moreover, this type of system is often reported by real users as unnatural and tedious in use.

2.3.2 Grammar rules generation

The second approach to be described here is the generation of grammar rules, which is used in many NLG systems.

NLG architecture here is represented as a pipeline of three stages:

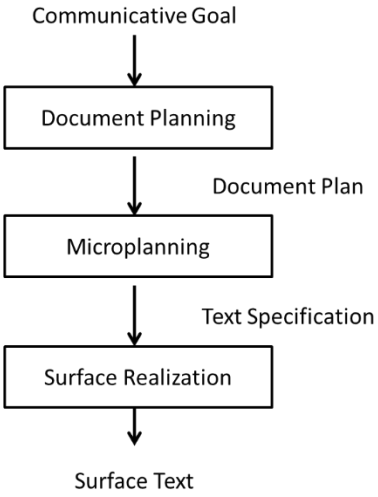

Figure 4. General NLG architecture [14]

Document planning determines the content and the structure of the document. This task includes constructing a set of messages from concepts and relations and is usually carried out by the dialogue manager.

Microplanning includes the process of lexicalization: determining which words will be used to represent the abstract concepts.

Surface realization converts the abstract representations into the actual text that expresses the desired meaning, applying syntactic and morphological rules.

However, one serious disadvantage of this method is that the generation of this grammar rules takes much effort and time, and only a highly skilled grammar writer can write a high-quality set of grammar rules. One advantage, though, is that most rules are domain independent, so once developed, they can be reused in other systems. One good example of such system is *SURGE* [15], which is still used as a component for many different applications. Another serious disadvantage is that the text used as input for this kind of models should be thoroughly parsed on the part of speech and additional information, which can be a big amount of work for medium and big-sized SDS.

2.3.3 Corpora-based approaches

Corpora-based techniques provide an alternative to rule and template-based techniques. It represents an attempt to capture the high quality of the language

used by a domain expert while minimizing the need for explicitly coding linguistic knowledge. The general idea is to use a corpus to automatically learn these rules of microplanning and/or surface realization [16]. The main advantage in comparison to the templates and grammar-rules approaches is that the natural ordering of words is implicitly encoded in the Language Models (LM), which are intended to capture statistical dependencies in word sequences. To learn those LM different statistical approaches are used.

A popular statistical approach is the one based on N-gram language model. N-gram language model consists of learning the frequency of n word sequences in a provided training corpus. It allows using the previous N-1 words to predict the next word. A hybrid template-based model augmented by attribute n-gram language model was implemented by [16]. All the utterances of the training corpus were divided into classes and then, a separate language model was built for each utterance class. A 5-gram model was used to balance variability in the output utterances with the objective to minimize the generation of nonsense utterances.

2.4 NLG evaluation

Evaluation of NLG could be difficult for various reasons. First, the NLG process cannot be evaluated out of the system context. The performance of the NLG components depends on language understanding and dialogue management task. Thus, the evaluation of NLG for a dialogue system can have many distinct objectives and consider many different dimensions of the system.

The main approach to the evaluation techniques, apart from the application potential evaluation and theory properties, are focused on properties of the generation system such as coverage, speech, correctness, etc.

There are several evaluation technics that can be divided into two large groups of automated and manual evaluation.

2.4.1 Manual evaluation

The most common methods of human evaluation are: user surveys and comparison of human-generated and system-generated output. These have the goal of scoring it based on one or several of the following metrics [17].

Accuracy is the measure of the extent to which the generated text conveys the desired meaning to the user. It can be summed up as “How easy is this sentence to be understood?” It requires an assessment between the input and output. It can be evaluated by comparing the system output with the expert output for the same test input.

Fluency metric consists in evaluating a quality of a generated text, which includes a syntactic correctness, organization of words in a sentence, and semantic coherence. It is an important metric due to some system need to produce an attractive output as well as functional. It can be summed up as “How well-written is this sentence?” To score the fluency, applying a human evaluation, several measurements can be used, such as sentence comprehension time or the time taken to post-edit the text. The most common approach is the direct human evaluation of the fluency, using questionnaires and specially developed rating systems.

Task evaluation involves observing how well a (possibly contrived) task is performed using the NLG system. It depends strongly on a system application domain and also on the user actions.

Oh and Rudnicky [16] describe an approach where their system is evaluated by running two versions of the so called CMU Communicator system, that differ only in whether the generation module is template-based or stochastic. Twelve subjects had one dialogue with each system, and then answered a survey and rated each system utterance on a scale from 1 to 3.

2.4.2 Automatic evaluation

As there is no “perfect sentence” to express a communication goal and there are many correct forms for one phrase, an automatic evaluation can be a difficult task.

A quantitative automatic approach is usually based on n-gram and word ordering. The most common measurement is Simple String Accuracy. It uses an ideal string output and compares it to a generated string using a metric that combines three Word error counts: insertion, deletion, and substitution

For the corpus-based dialogue system a tree-based automated method was proposed [18] where the generated output is compared to the original corpus, based on the same input. The tree-based accuracy metrics do not compare two strings directly, but instead build a dependency tree for the ideal string and attempt to create the same dependency tree for the generated string.

Another automatic evaluation metric is the *Bleu Metric* from IBM [19] which was initially used for machine translation evaluation. It takes n-gram appearance from multiple perfect sentences and then scores the generated output based on this statistic.

For the generation based on a statistical LM, it is possible to use the perplexity as an evaluation metric. Perplexity calculation can give us an idea about “the branching factor”. Applying to a LM it is calculated as the inverse of the average probability assigned to each word in the test set $(w_1 \cdots w_L)$ by the model [20]:

$$PP = \frac{1}{\sqrt[L]{\prod_{i=1}^{L} p(w_i)}}$$

The perplexity reaches its maximum when all the words have equal probabilities to occur. Appling to the NLG, we can calculate a perplexity of a generated result, using the generated corpus as a test data.

Anyway, the mentioned scoring metrics are still dependent on the syntactic structure, and human evaluation is considered as a more objective, though more expensive, evaluation method.

3. OBJECTIVES

The aim of this work is to generate a quality dialog output using provided corpus and a multiple-level language model train system. In previous work the here presented model was used for a Natural Language Recognizing task [21] and showed a better performance than the one-level LMs.

Since the word joints imply a higher amount of training information, the data problem can become highly serious. Therefore, combining phrase-based with class models can help to overcome this difficulty. This can be very useful for this kind of problems where obtaining a full and goal-specified corpus is a resource-demanding task.

4. APPROACH

4.1 Statistic language modeling

Statistical language modeling is an approach which tries to capture language regularities by learning the frequency of word sequences from a training corpus [22]. Typically LMs are based on n-grams, but some alternative approaches exist, such as LMs based on phrases or classes of words.

4.1.1 N-gram language models

The n-gram LMs are widely used in NLP, both for speech recognition and simple generation tasks. They estimate a probability of a symbol based on previous n-1 symbols.

A probability of a sequence of M words is calculated in this model as a product of conditional probabilities:

$$P(w_1 \dots w_M) = \prod_{i=1}^{M} P(w_i \left| w_1^{i-1} \right.)$$

The main problem of the n-gram LM is the data dispersion derived from irregular distribution of lexical units in the language. To solve this problem several softening techniques are used, such as lineal interpolation or recursive back-off to the lower order n-gram LMs.

Phrase n-gram LM could be considered as another approximation of n-gram models. The main idea of this approach is that the short phrase is used as the basic lexical unit instead of a word. For example, instead of count “buenos días” as two separate words, it is joined into a one lexical unit “buenos-días”. Nevertheless, the arbitrary assignation of phrases can provoke a greater dispersion of data in the training corpus. [23]

Another way to overcome the lack of training data in a specific domain is a class n-gram LM. To build such a model first the training corpus must be divided into several sets of words basing on some common characteristic.

4.1.2 K-TSS language model

This work uses the *k-TSS* (k-Testable in the Strict Sense) language models developed by [22]. K-TSS models can be acquired from training samples using an inference algorithm [24]. A usage of k-TSS grammars let us obtain a finite-state automaton associated with k-TSS model. This automaton determines a probability for every combination of k words, although it's impossible to obtain it all from a training sample.

A k-TSS language can be described by the following regular expression [25]:

$$L_{k-EE}\left(\Sigma, I_k, F_k, T_k\right) \cong (I_k\Sigma^* \cap \Sigma^* F_k) - \Sigma^* T_k \Sigma^*$$

where Σ is the alphabet, I_k is a set of initial symbols, F_k is a set of final symbols and T_k is a set of prohibited symbols.

The finite-state automaton extracted from a training set is defined as a quintuple $(\Sigma, Q^k, \delta^k, q_0, F)$ where in this case

- $\Sigma = \left\{w_j\right\}, j = 1..\left|\Sigma\right|$ is the vocabulary, or the set of all the words in a training corpus

- Q^k is the set of states of the automaton, associated with the k-level model. Each state is a sequence of words

- F is the set of final states of automaton

- δ^k is the transition function which defines the destination state $q_d \in Q^k$ and the probability $P(w_i / q) \in \left[0\ldots1\right]$ of the transition from a state q into the word w_i Every transition between two states is a sequence of k words, where the original state is the sequence of first (k-1) words and the destination state is the sequence of the last (k-1) words

The probability of every transition is estimated by maximum verisimilitude criteria as follows:

$$P(w_i / w_{i-(k-1)}^{i-1}) = \frac{N(w_i / w_{i-(k-1)}^{i-1})}{\sum_{\forall w_j \in \Sigma} N(w_j / w_{i-(k-1)}^{i-1})}$$

One of the common problems while constructing a LM is data sparsity. Therefore softening techniques are used to subtract probabilities of combinations and distribute them among the combinations which have not appeared in the training sample. In the k-TSS models used in this work, the softening techniques known as "back-off" were applied. This is possible by the use of a recursive model which integrates K models in one.

For a back-off transition $P(w/b_q)$ is the probability associated to the same event in the submodel (k-1)-TSS.

The finite net obtained by the method proposed by Varona is represented as an array. Every state is represented by $\left|\sum_q\right|+1$ positions, where every position represents a possible transition. Each position of the array represents a pair (q,w), where $q \in Q^k$ and $w \in \Sigma_q \cup \{U\}$, and consists of four elements:

- A value of $\left|\sum_q\right|$, a number of different events in every state
- A transition for each word $w \in \Sigma_q$ or a (-1) state, which designates any event not listed previously $w_i \notin \Sigma_q$
- A probability $P(w_i / q) \forall w_i$ or $P(b_q / q) \forall w_i \notin \Sigma_q$
- A link to the first node of q or its back-off state b_q

The following example shows a part of an array which represents a model trained from DIHANA corpus (user speech part) with K=3. This corpus will be addressed in Section 5.

Event number	**Word number in vocabulary file**	**P (w/q)**	**Destination**

866	0	0.111079560121844	867
866	187	0.062664567091745	1089
866	257	0.052609840466725	1271
866	2	0.037598740255047	1379
…	…	…	…
866	-1	0.000000000000000	0
221	729	0.312677013707942	8710
221	506	0.162342811827348	8834
221	257	0.054491899852725	8893
221	620	0.051999546844908	8948
…	…	…	…
221	-1	0.095366400064877	0

Table 3 Part of a trained model table for DIHANA corpus

The complete table consists of $\left|Q^k\right|\times(\left|\Sigma_q\right|+1)$ positions, where every state is represented by $\left|\Sigma_q\right|+1$ positions, each position representing a transition. The last position in every state represents a back-off state and a probability associated with a back-off transition.

4.1.3 Hierarchical language model

The described LM were adapted by Justo in [25] in order to build a two-level hierarchical language model (HLM), based on words classes, being the main goal to deal with data sparseness. The resulted HLM then was implemented as a part of Automatic Speech Recognition (ASR) system.

To implement a hierarchical model, first of all a classical word k-TSS LM was considered (M_w), where the probability of a sequence of N words, is obtained considering the history of previous $k-1$ words.

Then two different approaches for HLMs (M_{sw}, M_{sl}) were considered. In the first approach, M_{sw}, a set of classes made up of phrases constituted by not linked words is used. In this way, the probability of a word sequence ($\overline{w}$) can be computed as follows where the segmentation (s) and classification ($\overline{c}$) of a word sequence are considered as hidden variables:

$$P(\overline{w}) = \sum_{\forall \overline{c} \in \Sigma_c^*} \sum_{\forall s \in S(\overline{w})} P(\overline{w}, \overline{c}, s) = \sum_{\forall \overline{c} \in \Sigma_c^*} \sum_{\forall s \in S(\overline{w})} P(\overline{w} \left| s, \overline{c}\right.) P(s \left| \overline{c}\right.) P(\overline{c})$$

In the second approach, M_{sl} , classes are made up of phrases constituted by linked words, $\bar{l}$. Thus, the probability of $\bar{w}$ is given as:

$$P(\bar{w}) = \sum_{\forall \bar{c} \in \Sigma_c^*} \sum_{\forall \bar{l} \in \Sigma_l^*} P(\bar{w}, \bar{c}, \bar{l}) = \sum_{\forall \bar{c} \in \Sigma_c^*} \sum_{\forall s \in \Sigma_l^*} P(\bar{w} \left| \bar{l}, \bar{c} \right.) P(\bar{l} \left| \bar{c} \right.) P(\bar{c})$$

where Σ_c^* is the set comprising all possible class sequences for the given Σ_c alphabet of classes and Σ_l^* is the set of all possible sequences of l_i phrases.

The final model M_{sw} is not only one automaton associated to each LM but $N_c + 1$ different SFSA (Stochastic Finite State Automaton) are needed, where N_c is the size of the set of classes: one for each class considering the relations among words inside the classes and an additional one that takes into account the relations among classes.

For integration of the different SFSA in the ASR system a dynamic composition was carried out, the different models are integrated into the search network for a classical k-TSS LM.

The following example to illustrate is proposed [25]., using the M_{sw} model. Being $\Sigma_c = \{c_1, c_2\}$ a two-class vocabulary made up of phrases, where $c_1 = \{w_1, w_1\, w_1\}$ and $c_2 = \{w_2\, w_3, w_1\, w_2\, w_3, w_4\}$. Figure 5 represents the automata that take into account the relations among classes, and Figure 6 represents the specific automaton for the class c_2. When considering M_{sl} model, the SFSA associated to each table is a 1-TSS model.

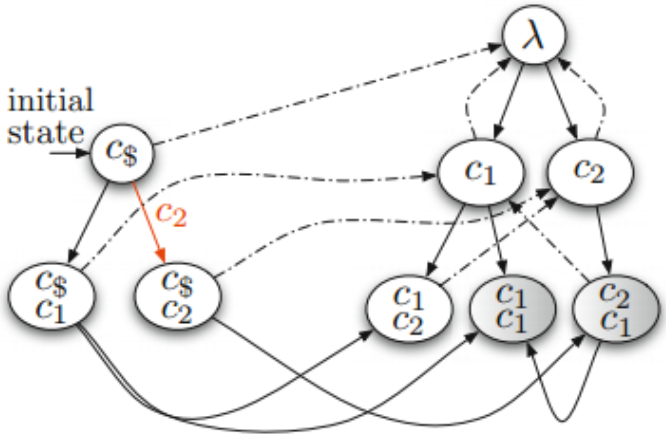

Figure 5 SFSA that illustrates the relations among classes [25]

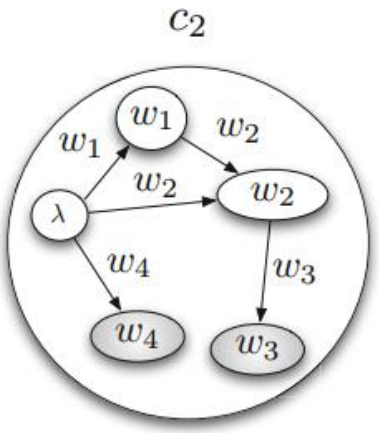

Figure 6 SFSA for a c_2 class [25]

5. EXPERIMENTAL DESIGN

5.1 Corpus used in training

Two domain-specific Spanish corpora were selected. One of them is the Bilbobus (010) corpus, which consists of 197 recorded dialogues in Spanish between real users and operators of informational system of a bus company. The obtained records were converted into text files, retaining all kind of utterances, including pauses and confirmations. Every dialogue was manually processed, describing its specific dialogue goal, language, gender of user, dialogue acts of user and operator.

```
O: Bien, vale, vale. A ver, el setenta y siete me ha dicho ¿no?
[confirm_bus_line]
<confirm_yes><bus_line><ask_confirm>
```

Figure 7. An example of Bilbobus (010) corpus tagging

Bilbobus corpus was tagged within a two-level tag system. First level contains the main sentence goal and the second level separates a sentence into a set of different speech acts.

```
<confirm_yes>
<bye>
<out_of_domain>
<depart_time>
<greeting>
<bus_line>
<ask_confirm>
<agent_name>
<ask_wait>
<frequency>
<out_of_task>
<depart_place>
<depart_neighborhood>
<acknowledge>
<down>
<arrive_place>
<processing>
<both>
<no_alternative>
<twenty_to>
<place>
<nearby>
<youwelcome>
<from_time>
<schedule>
<ask_schedule>
<time>
<part>
<ask_bus_line>
<confirm_no>
<confirm_return>
<tomorrow>
<after_arrive>
<twenty-five_to>
<start_date>
<same>
<now>
<bring_closer_little>
<ask_tram>
<date>
<ask_frequency>
<arrive_neighborhood>
<interval_time>
<neighborhood>
<ask_change_now>
<tell_me>
<ask_depart_neighborhood>
<complicated>
<all_month>
<twenty_past>
<ask_depart_time>
<position>
<ask_day>
<ask_interval_frequency>
<return>
<when>
<not_save>
<reference>
<depart_place_part>
<sorry>
<ask_schedule_change>
<bus_stop>
<messy>
```

Figure 8 The set of tags for the second level of semantic tagging of Bilbobus corpus

The second corpus is a spontaneous-speech dialogue corpus named DIHANA. This corpus was acquired by the Spanish universities community by DIHANA project, which was aimed to construct a dialogue system for consulting of Spanish nation-wide trains [26]. It was recorded using the *Wizard of Oz* technique, when a behavior of a system is simulated by a real person. The task consisted of the retrieval information on Spanish nationwide trains obtained by telephone. In order to obtain more realistic dialogues, the speakers had to reach a certain goal in each dialogue: get one way schedule, get one way price and schedule and get roundtrip price and schedule.

First level	Afirmacion, Apertura, Cierre, Confirmacion, Espera, Indefinida, Negacion, No_entendido, Nueva_consulta, Pregunta, Respuesta
Second level	:cortesia :consulta :coletilla :m_salida :m_llegada :<hora> :<hora_llegada> :<duracion> :fecha :ciudad_destino :ciudad_origen :tipo_tren :tipo_viaje :<precio> :numero_relativo_orden :clase_billete :<afirmacion> :<negacion> :nada
Third level	(AFIRMACION), (DURACION), (HORA) (HORA-LLEGADA), (HORA-SALIDA) (NEGACION), (NO-ENTENDIDO) (PRECIO), (SERVICIOS) (TIPO-TREN)

Table 4 Third level tagging scheme in DIHANA corpus

The DIHANA corpus was tagged in a three-leveled scheme [27]. The first level serves for tagging the whole dialogue turn and represents its intention.

The second level separates a dialogue act into different semantic segments. Third level serves to store specific information, for example, the name of a city destination. The Nil level was used to annotate levels without an undefined value as it can be seen in Figure 9

```
U0019: hola buenos d'ias mira quer'ia saber horarios de trenes
para ir a ciudad_real \
hola buenos d'ias:cortesia\
mira quer'ia saber:consulta\
horarios de trenes para ir:<hora>\
a ciudad_real:ciudad_destino\
TIPO-VIAJE:nil\
(HORA)\
CIUDAD-DESTINO:ciudad_real\
```

Figure 9. An example of DIHANA corpus tagging

5.2 Generation software

The generation software was written in Python programming language. As input it receives a trained LM and a vocabulary, which contains all of the words used in the training set. The output is a phrase in natural language.

The generation function is the recursive function which is implemented based on the model structure. To generate the sequence of states the generation function takes as input the LM represented as a FSA table, a number of words to generate and the vocabulary size of the model. It performs a loop in order to select N phrases to be generated. Within every loop a word (or a class in a class-based model) is inserted into a sequence of word tags.

A generated sequence starts with the second state of the automaton. This state represents a set of all possible starting words of a model. A transition function chooses the transition to be done and the word associated to this transition.

The word tag, which is allocated in the second column of the FSA table, is appended to the output sequence. Then, q takes the value from the fourth column, which is the first node of the next state of the automaton.

```
def transitionFunction(q, model):
    "Define a transition function"
    r ← random.random(0,1)
    P ← 0
    while P < r:
        P ← P + model[2][q]
        q ← q + 1

    return (q)
```

```
def generateSequence(N, words,
model):
    "Generate a sequence of states
    N - number of phrases to generate
    words - number of words in
dictionary
    model - FSA table"
    q ← words #initial state
    phrase ← []
    i ← 0
    while i < N:
        aux ← transFun(q, model)
        phrase.append(model[1][aux])
        q ← model[3][aux]
        if model[1][aux] == 0:
            i ← i + 1

    return(phrase)
```

The transition function selects the next state given the actual state of the automata following the approach of [28]. It takes as input the first position of the current state q and the FSA table. A random value of r is generated on every call in the interval (0, 1] starting a cycle that sums up the probabilities, going down the FSA table. The transition function returns the number of the transition where the summed transition probability exceeds the randomly obtained number, thereby the more probably transition allocated on the first part of the table has more possibilities to be chosen.

For the class-based model the process of generation is recursive. First the sequence of states is generated while using as the input a FSA table trained on the corpus of classes (presented further on Figure 12). Then for each class of the sequence the process is repeated with the input model M_{cw} inside each class.

5.3 Model structure

To generate a k-TSS LM based on classes only the second level of tagging is used. Additionally, the class represented by the tag “consulta” is divided into several classes, depending on the phrase structure. First, it is divided into two big sets, the ones which start with “and” and could be located only in the

middle of one phrase, and the ones which start with any other word. Then, both are divided into various separate subclasses depending on the final word of the phrase. The final division is explained in Table 5:

Clases	***Subclases***	***Example***
Without "and"	ends with "de" ends with "que" ends with "en" ends with "ir" ends with "es" ends with "tren/trenes" ends with "ser/sea" ends with "viaje" ends with "viajar" the rest of frases	me gustaria un viaje de me puedes decir el que querria hacerlo en queria informacion para ir querria saber cual es quiero informacion de trenes quiero que el billete sea tendria que ser el viaje pues quiero viajar querria conocer tambien
With "and"	ends with "ir" ends with "es" ends with "tren/trenes" ends with "ser/sea" the rest of frases	y si es posible ir y digame cual es y a ser posible en un tren y si pudiera ser y tambien quiero saber

Table 5 A division example of the class tagged as "consulta"

6. EXPERIMENTS

The corpus described in previous section is divided into user speech and system speech parts and the generation is made separately for each of the groups and for the combination of both corpora. The main reason for this division is that the used vocabulary and tags sets are different for the user speech acts and the system speech acts. It is related with the difference between the speech goal of each subject of the dialogue. Table 6 describes some characteristic of used corpora. For n-gram models the size of vocabulary is indicated, and for class-based modes also the size of the set of classes. The combination model for n-grams is constructed as the union of both corpora. The combination model for classes is described in Section 6.2

	Bilbobus (010)		**DIHANA**		**Combination**	
	System	*User*	*System*	*User*	*System*	*User*
Word n-gram LM	1867 words	2026 words	351 words	865 words	2116 words	2424 words
Statistical classes LM	X	X	X	300 classes	X	X
Semantic classes LM	X	X	X	35 classes	X	38 classes

Table 6 Properties of each model

6.1 Generation based on n-gram models

For each part of the corpus several models were generated with different values of K, from K=2 to K=5. As input parameters, a trained model in form of

a FSA table and a sorted dictionary of the model are taken. The phrase generation stops when the generation algorithm reaches one of the final states of the associated automaton. Tables below present the generation result for Bilbobus, DIHANA and combination of both corpora.

	System turns, K = 2
Bilbobus	muy bien bien veinte minutos exactos a ver cincuenta siete y no decia y cuarto . por favor un poquito a ver . si claro se recorre el horario de acuerdo . enfrente de . 010 arratzalde on .
DIHANA	le consulto horarios de trenes regionales y llega a cadiz el lunes uno de junio un intercity sale a las veinte de informacion . le consulto horarios de ocho de octubre no el de largo recorrido por utilizar este servicio solo puede realizar otra consulta . hay siete horas y vuelta algo mas . hoy mismo . bienvenido al sistema realizar el de trenes euromed a otra consulta.
Combination	de las trece treinta y diez minutos dispone de la que sale a ver pues deusto . uno eso ya tiene que puede y vuelta algo mas . desde la manana el tren dispone de zaragoza a usted agur . que le atiende paola . puede coger .

Table 7 Generation results for system turns, K=2

	System turns, K = 3
Bilbobus	diez y menos cuarto . antes del arenal san inazio sale cada veinte minutos a y media y menos veinticinco . agur egun on le atiende alicia . bueno el treinta y ocho que viene de la ponen la asociacion de comerciantes del casco viejo . en invierno .
DIHANA	. el precio del viaje de ida en segunda clase en un momento por favor . quiere ir de burgos a san_sebastian el jueves dieciseis de abril un momento por favor . un momento por favor . el precio del viaje de ida y vuelta algo mas . ese tren es un ave .
Combination	un momento por favor utilice frases cortas que desea . trenes el primero sale a las quince horas y cincuenta y cinco minutos de la manana desde termibus va para en un momento . si . si del muerto si a las quince horas y quince minutos y el tercero a las diecisiete horas y cuarenta

	minutos y llega a las diecinueve horas de la ria . entre las seis y cuarto a y veinte minutos salida de larraskitu cada quince minutos tiene parada .

Table 8 Generation results for system turns, K=3

	System turns, K = 4
Bilbobus	cero diez egun on le atiende itziar. eso es y a y a y diez y a media . bilbobus permite perros domesticos y hay hay eh o sea la la principal la que une vamos a ver . tiene a las once y luego ya ambulatorio desde el dieciocho . desde barakaldo y media diez y media para coger cual el en el puente del arenal .
DIHANA	ese tren es un intercity algo mas . un momento por favor . a oviedo el martes uno de agosto un momento por favor . hay dos trenes el primero sale a las ocho horas y las trece horas hay dos trenes el primero sale a las doce horas y cincuenta y cinco minutos desea realizar otra consulta . ese dia hay tres trenes el primero sale a las nueve horas y el segundo a las diez horas y cincuenta minutosde la manana desea algo mas.
Combination	de artzanda a deusto . muy bien a gijon el domingo nueve de abril un momento . pero luego a la vuelta de la manana y el ultimo a las veintidos horas y quince minutos algo mas . a usted por llamar agur . claro lo que estoy viendo entonces algun si plaza algo mas .

Table 9 Generation results for system turns, K=4

	System turns, K = 5
Bilbobus	un momentito por favor sobre que hora queria . no encuentro nada a usted hssta luego . vamos eso es . de acuerdo . un poquito mas bien indautxu tiene tambien el treinta y seis que es de donde sale de moyua no hay nada .
DIHANA	el precio del viaje de ida y vuelta desea realizar otra consulta .

	ese dia hay catorce trenes el primero sale a las ocho horas y cuarenta minutos de la manana desea mas informacion . lo siento pero creo que no le he entendido bien de zaragoza a león . el tren que sale a las dieciseis horas y diecinueve minutos desea algo mas . hay un tren que sale a las doce horas y quince minutos de la manana y el ultimo a las once horas y cuarenta y cinco minutos algo mas
Combination	madrid a cordoba el sabado treinta de octubre . de siete mil pesetas el de ida y vuelta desea realizar otra consulta . y si el veintidos espere un momentito ah de sarrko espereme . le consulto horarios de zaragoza a barcelona el jueves siete de octubre un momento por favor . desde que zona .

Table 10 Generation results for system turns, K=5

	User turns, K = 2
Bilbobus	muy bien bien vale vale de moyua y el numero . siete de que coge el trayecto es el autobus tengo apuntao yo no desde cinco y a necesitar . si vale vale . el cero y el favor que hora . es por por vale vale .
DIHANA	bueno querria un poco mas muchas gracias . si me gustaria conocer el dia a tarragona me gustaria saber si puede a vitoria . quiero hacer el tren alaris P los precios . si el precio . salir un viaje del billete los horarios y llegada ha dado desde alicante .
Combination	hola si mire la mina del mediodia de abril . quisiera trenes P obtener el si quiero salir por la otra punta . si . tren P no despues de la tarde . semana por favor .

Table 11 Generation results for user turns, K=2

	User turns, K = 3
Bilbobus	si . vale gracias eh . de las oficinas del bilbobus .

	vale pues muchas gracias . setenta y seis si si es cada doce minutos .
DIHANA	querria ir de bilbao a madrid antes de las dos de las seis de abril . querria llegar el viernes nueve de abril . hola buenos dias P queria saber horario de un viaje a valladolid pasado manana . el proximo viernes P pero que van de valencia hasta el viernes que viene . no .
Combination	hombre a santander el proximo dia dieciseis de abril . ni la tarde . no tengo boli aproximadamente proximo sabado . los horarios de ida de valencia a barcelona manana por la tarde . hola mira querria saber horarios y precio P del todo a pamplona y el precio de un viaje de valencia a teruel .

Table 12 Generation results for user turns, K=3

	User turns, K = 4
Bilbobus	en que que horarios cada cuarto de hora . va por hurtado de menos diez eh en algun sitio bueno . agur . vale pues muchisimas gracias perdon por todo eh . a ver queria a ver si subo arriba no se donde .
DIHANA	hola buenas tardes me gustaria saber el precio P el doce y el catorce de mayo por favor . hola buenas tardes queria queria saber malaga no quiero llegar a avila . si P quiero llegar antes de las diez y las diez o a las once y cinco llega antes de las cinco . quiero ir a badajoz a cualquier a las diez de la manana en un tren regional
Combination	hola buenos dias quisiera viajar a alicante . bien P y la llegada antes de las doce de mayo . si para la persona que la vuelta sea el dia diecisiete de septiembre . si . querria saber horarios y precios .

Table 13 Generation results for user turns, K=4

	User turns, K = 5
Bilbobus	a ver que frecuencias tiene mas o menos y se coge si le pilla cerca o le pilla lejisimos porque baja por hurtado amezaga . aupa arrstsalde on mira me puedes usted iria desde donde me ha dicho a ver . a ver a que a que horas pasa por indautxu y que me has dicho que hora sale de san inazio . seis menos el hospital que han cambiau tanto las las lineas . adios agur
DIHANA	si me parece muy bien gracias . el martes que viene por la manana . no es salida al mediodia salida a la tarde y el tipo de tren . queria saber los horarios de trenes desde zaragoza a logrono los domingos . si me gustaria saber los horarios de trenes para ir a madrid manana .
Combination	no quiero ir a valencia . muy bien fenomenal fenomenal pues vale venga gracias . si me gustaria ir en un tren P de tipo talgo . si deseo saber que tipo de tren es el triana . si .

Table 14 Generation results for user turns, K=5

6.2 Generation based on class models

Three different class models were used to build class-based LMs. First model represents a statistical classification, where the phrases were grouped into classes based into the frequencies of its occurrences in the training corpus.

```
a_qué_hora_llega
cuánto_cuesta
efectivamente
exactamente
qué_precio_tiene
qué_tipo_de_tren_es
```

Figure 10 Example: statistical class #174

The second model is based on semantic tagging of the user turns of corpus DIHANA as described in Section 5. Each class contains the phrases tagged

in the training corpus as belonging to the set. On Figure 11 a class extracted from “Afirmación” tag is represented.

```
vale pues                       esta bien
si quiero                       eso si
si que quiero                   eso seria perfecto
si pues                         eso quiero
si eso es                       eso es lo que quiero
si asi es                       eso es
quiero                          en efecto
que si                          efectivamente
pues si                         de acuerdo
por supuesto                    creo que si
perfecto                        claro
necesito                        bueno
muy bien                        bien si
me viene bien                   bien estaria bien
me va bien                      bien esta bien
me gustaria                     bien
exacto                          asi es
exactamente                     si
```

Figure 11 Example: Class Afirmación

The third class-based LM is constructed with the user turns of both corpora, DIHANA and Bilbobus. Class “consulta” from the previous model was enhanced with expressions extracted from Bilbobus corpus and then divided into various subclasses. Only the domain-independent phrases were extracted, tagged by the following tags:

```
<ask_bus_line> <ask_frequency> <ask_depart_time>
<ask_interval_frequency>
```

6.2.1 Two-level generation

For a two-level generation a specific LM is needed as well, which is built by tagging each phrase of the corpus based on the class it belongs to. The sample of this model is presented by Figure 12.

```
hola mira que queria ir a malaga desde aqui desde bilbao y queria
saber los horarios de los trenes $
no $
a malaga $
```

```
el sabado por la tarde $

clase35 clase34 clase12 clase24 clase34 clase25 clase6 clase22 $
clase33 $
clase24 $
clase23 clase19 $
```

Figure 12 A sample of class tagging corpus

Based on this model, first, a random sequence of classes is generated in the same way that it was used for the n-gram generation. Each class has its own LM which evaluates the probability of each phrase of a determined class in the training corpus. Thus, the generation is recursive: the construction of a phrase inside every class repeats the construction process of the whole phrase.

K = 3	quiero cual_es_el_precio_del euromedia deseo saber_horarios_para_ir_a ir_a ciudad_real desde_madrid si de gijon P_a granada P_el_dia diez las_diez_y las_diez_y media entre_las tres si obtener_el_horario P_para_el viaje de almeria este_viernes P cual me puede decir para viajar_a valladolid este_viernes si_por_favor por_favor

Table 15 Generation based on statistic classes

In a statistical-based class model, the phrases within a class are considered as single units. Therefore, the low-level model inside the class works as simple 1-gram model and the transition function serves as a random phrase-generator.

In a semantic-based class model, it is possible to choose the K parameter while training a low-level model within the class. In Table 16, the results of generation of the second model with the different K parameters are presented. The generation was based on the user speech part of the DIHANA corpus.

	K = 2 (low-level model)	K = 3 (low-level model)
K = 2	queria saber si los horarios de para ir . el domingo . no nada mas gracias . quiero saber que es talgo de la tarde antes . muchas gracias el horario me gustaria viajar en un tren que y llegada de la tarde .	para ir a madrid si las cinco el tren alaris horarios a la_coruna si quiero querria obtener horarios el precio si viernes por la tarde y bueno pues ida el viaje de ida de valencia
K = 3	quiero es posible precio . si que el tren sea si quiero . quiero viajar en el euromed para manana el sabado . si que saber precio del billete para finales de marzo sabado . tipo de tren .	si es quisiera obtener horarios saber el precio ida ida el precio nada mas gracias . queria si es posible a valencia jueves el domingo . informacion salir o martes dia once a ver llamaba para obtener el tipo de tren para salir de la manana . esta bien . el once de enero el proximo viernes .
K = 4	queria un viaje de viaje de ida y vuelta a santander este sabado . el proximo viernes . no nada gracias . horarios de para ir a pamplona si salir en un tren rapido . si .	si un viaje de ida de ida . no el horario desde santander a sevilla manana . no . los horarios mira queria ir a barcelona . gracias horario pues quiero mirar ir a caceres el dia veinticuatro de mayo por la manana mas rápido .
K = 5	si el horario para viajar rapido a granada . los horarios a madrid a cordoba viernes ave queria saber . no no deseo nada mas muchas gracias . quiero un viaje de ida a castellon destino tarragona me gustaria ir proximo viernes volver el catorce . el dieciocho de .	quiero obtener horarios del viaje salir de castellon a teruel me gustaria querria que el tipo de tren antes a sevilla querria saber horarios para un viaje nada mas a ver nada para ir a bilbao pues querria saber precio un viaje vuelta a logrono a logrono

Table 16 Generation based on semantic classes

To see the generation of the third model combined from both corpora, it's precise to choose the sequence of classes where the enhanced classes are used. Therefore the results for this generation are presented only in Section 6.2.2

6.2.2 Generation with a fixed class sequence

In the real-life usage of the generation system, the system will not generate a random phrase, but instead, it will receive from a dialogue manager a specific sequence of classes that need to be converted into a phrase in natural language, understandable by a user. The third part of the experiments is made taking into account this consideration.

As input to a generation program a sequence of classes corresponding to a phrase which already exists in the training corpus is taken. In tables below the original phrases taken from the training corpus, the sequence of classes associated with the phrase and four output phrases generated based on this sequence are presented.

Original phrase	**queria ir a la_coruna y queria saber los horarios que hay de los trenes para el lunes $** **clase12 clase24 clase6 clase22 clase23 $**
Generated results	para ir a lerida y queria el horario para ir el veinte de marzo me gustaria ir a huesca y si es posible el horario lunes quince de marzo que quiero ir a barcelona y querria saber horario el diecinueve de diciembre quisiera ir a sevilla y queria horarios de sabado doce de abril
Original phrase	**el tren que sale a las tres treinta a que hora llega a oviedo** **clase8 clase15 clase19 clase22 clase18 clase24**
Generated	queria buscar un tren que sale antes y el horario para llegar a valladolid

results	hay algun tren que salga entre las ocho de la tarde el horario de para llegar a granada algun tren que sale a partir de las tres y media horarios llegar para gijon hay algun tren que salir antes de las veintitres treinta que horarios hay para llegar a barcelona
Original phrase	**hola buenas me gustaria obtener informacion sobre un viaje de ida a madrid para el sabado** **clase35 clase4 clase28 clase24 clase23**
Generated results	saber queria un billete de de ida a bilbao veinticinco de junio hola buenos dias un viaje de un billete de ida con destino madrid viernes saber del tren de ida a teruel el dia quince me gustaria deseaba de solo ida a barcelona el viernes que fuera volver a la coruna domingo

Table 17 Existing phrase based generation for the second model

To see results of generation for the combined model, it's precise to choose the phrases with the enhanced classes. The following tables contain four phrases generated with the third model.

Original phrase	**si me podria decir los horarios del sabado por la manana** clase10 clase22 clase23 clase19
Generated results	oye una pregunta horario para ir miercoles a partir de las siete querria saber horario jueves de la mañana a ver mira me puede decir horario de catorce de mayo a las cuatro de las siete de las cinco queria saber el horario el viernes a partir de las cuatro
Original	**si me podria decir el precio del billete** clase10 clase29

phrase	
Generated results	oye mira que hora sale y los precios oye una preguntita queria saber los precios querria saber precio me gustaria saber el precio
Original phrase	**queria ir a tarragona y queria saber los horarios de los trenes** clase13 clase25 clase10 clase22
Generated results	queria informacion para ir a ciudad_real vamos a ver si queria saber horario de quiero ir sevilla me puede indicar horarios de me gustaria viajar a vigo queria saber los horarios de trenes mira quiero viajar a madrid me gustaria saber a que hora

Table 18 Existing phrase based generation for the combined model

6.3 Evaluation of results

In the present work, when choosing the appropriate generation metrics, the fact that the system work was simulated by passing a sequence of categories directly into a surface generator should be taken into account. Therefore, the task evaluation cannot be made for the n-gram models, where the phrases are randomly generated.

To evaluate the generation results for the n-gram LMs the perplexity was calculated for some combination of order and corpus used in model construction. To calculate the perplexity for each model an artificial corpus was generated, with a size of approximately 15.000 words. The overall results are presented on Figure 13 and Figure 14.

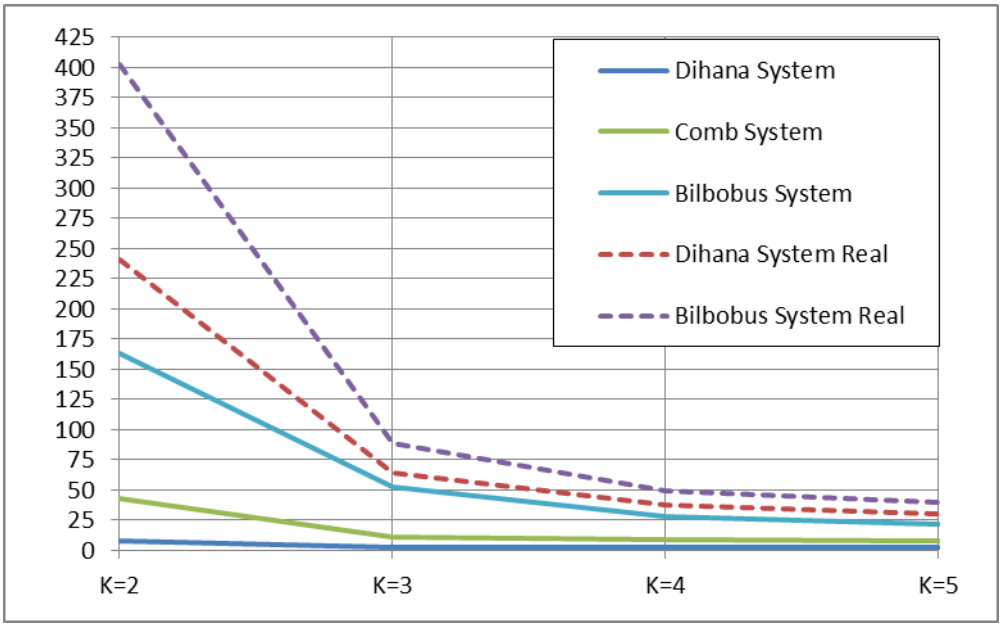

Figure 13 Perplexity calculated for the system turns of corpora

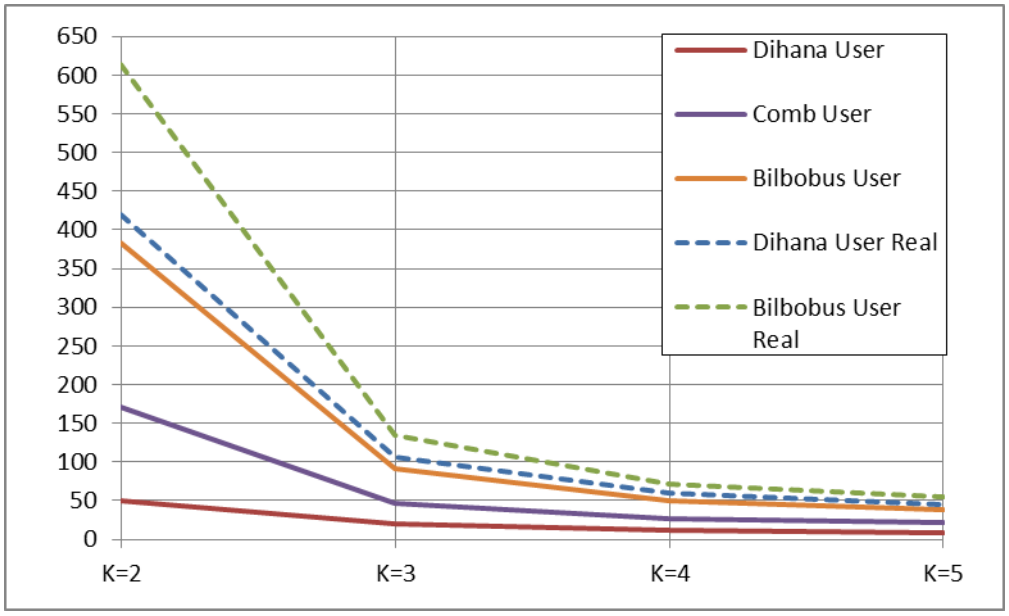

Figure 14 Perplexity calculated for the user turns of corpora

The perplexity values for the K=2 are much higher for the real corpora, but almost equals with the generated corpora for the K=4 and K=3 values. It means that for these values of K the generated corpora approach well to the real corpora.

The clear difference in perplexity magnitude between the system models and the user models is explained by the nature of corpora. DIHANA system turns were recorded as an automatic answer and have a clear structure and constantly repeating patterns, which significantly reduces the entropy. Whereas the user model for both corpora have much higher values of perplexity, this is explained by the fact that both corpora consist of spoken language, which tends to be full of repetitions, incomplete sentences, that increases the overall entropy.

For all models the difference in perplexity is sensibly reduced from 2-gram to 3-gram model. However, this difference reduction is almost insignificant from the 4-gram to the 5-gram one. This fact makes us conclude that for Spanish language generation model using the spoken corpus it is enough with a K value of 4. Increasing the K value doesn't provide a significant improvement for the model performance, but instead, increases a model complexity.

The evaluation of the class-based hierarchical model cannot be performed by automatic perplexity calculation. Although the overall vocabulary is the same for n-gram model and the class-model, the probabilities calculation in case of class-model is performed on an additional "tag" corpus (presented previously on Figure 12).

The BLEU metric [29], used in Machine Translation tasks cannot be applied either to the automatic evaluation, as it is based on the difference between the "perfect" translation and resulting sentence. In case of class-based model generation the difference between the perfect output (phrase from real corpus) and the generated one can be significant because of the way the classes were formed. For example, phrases such as "desde madrid", "desde zaragoza", "desde valladolid" from the class "ciudad_destino" are treated as equivalents in the generated output, while the BLEU score will count them as a wrong translation for the original phrase "desde bilbao"

One alternative would be to apply manual evaluation methods. These methods are based on questionnaires formulation, taking into account that the task evaluation can be realized only after the joint implementation with the dialogue manager, which is outside of the scope of present work.

7. CONCLUSIONS AND FUTURE WORK

In the present work a system of natural language generation based on statistical hierarchical k-TSS models is presented. The application domain is a dialogue system which realizes consults about transport timetables. Both corpora used for generation of LM are dialogues with real users in Spanish. The generation was made first using a simple n-gram LM, and then, using a hierarchical LM, where the high-level model was based on classes and the low-level model inside a class was based on phrases.

As future work, several modifications of the present system are proposed. The current NLG model has several limitations. The most significant one is that the vocabulary used in generation matches exactly the vocabulary of the training set. The possibility of including WordNet synonyms dictionary (synsets) can be considered as a way to increase the vocabulary and expressiveness on the stage of post-processing of the generated output.

Another limitation is the basic principle in which the classes were formed. In the current work, the classes were formed only with semantic information. Probably better results could be achieved when not only the semantic part is considered, but also the syntactic structure of the sentence.

Finally, as second stage of NLG, a grammar parser could be used in order to improve the quality of the raw output, eliminating such errors as the singular/plural choice, the usage of definite or indefinite articles and the gender suffixes concordance, which are very regular in Spanish language.

Regarding the evaluation metrics for class-based model, one of the possible solutions could be adjusting the existing automatic evaluation methods as *BLEU* for the output based on determined sequence. In some sense the process is similar to Machine Translation, the field where BLEU is mostly used, although a direct application of the metric gave unsatisfactory results.

After applying the listed improvements, the resulting system could be used as the input generator before for the speech-synthetizing stage in a SDS inside a transportation domain.

8. REFERENCES

[1] Jurafsky, Daniel, and James H. Martin. "Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition." (2000).

[2] McEnery, Tony. Corpus linguistics: An introduction. Edinburgh University Press, 2001.

[3] Schmid, Helmut. "Probabilistic part-of-speech tagging using decision trees."Proceedings of international conference on new methods in language processing. Vol. 12. 1994.

[4] Giménez, Jesús, and Lluis Marquez. "SVMTool: A general POS tagger generator based on Support Vector Machines." In Proceedings of the 4th International Conference on Language Resources and Evaluation. 2004.

[5] Dimitrova, L., Ide, N., Petkevic, V., Erjavec, T., Kaalep, H. J., & Tufis, D. (1998, August). Multext-east: Parallel and comparable corpora and lexicons for six central and eastern european languages. In Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1 (pp. 315-319). Association for Computational Linguistics.

[6] Miller, G. A. (1995). WordNet: a lexical database for English. Communications of the ACM, 38(11), 39-41.

[7] Atserias, J., Casas, B., Comelles, E., González, M., Padró, L., & Padró, M. (2006, May). FreeLing 1.3: Syntactic and semantic services in an open-source NLP library. In Proceedings of LREC (Vol. 6, pp. 48-55).

[8] Bohus, D., Raux, A., Harris, T. K., Eskenazi, M., & Rudnicky, A. I. (2007, April). Olympus: an open-source framework for conversational spoken

language interface research. In Proceedings of the workshop on bridging the gap: Academic and industrial research in dialog technologies (pp. 32-39). Association for Computational Linguistics.

[9] Taulé, Mariona, Maria Antònia Martí, and Marta Recasens. "AnCora: Multilevel Annotated Corpora for Catalan and Spanish." LREC. 2008.

[10] Bisbal, E., Molina, A., Moreno, L., Pla, F., Saiz-Noeda, M., & Sanchis, E. (2003). 3LB-SAT: Una herramienta de anotación semántica. Procesamiento del lenguaje natural, 31, 193-200.

[11] Castellón, I., Fernández-Montraveta, A., Vázquez, G., Alonso, L., & Capilla, J. (2006). The SENSEM corpus: a corpus annotated at the syntactic and semantic level. In 5th International Conference on Language Resources and Evaluation (LREC 2006).

[12] Smeele, Paula MT, Sebastian Möller, and Jan Felix Krebber. "Evaluation of the speech output of a smart-home system in a car environment." INTERSPEECH. 2004.

[13] Theune, Mariët. "Natural language generation for dialogue: system survey." (2003)

[14] Reiter, Ehud, Robert Dale, and Zhiwei Feng. Building natural language generation systems. Vol. 33. Cambridge: Cambridge university press, 2000.

[15] Elhadad, M., & Robin, J. (1996). An overview of SURGE: A reusable comprehensive syntactic realization component.

[16] Oh, A. H., & Rudnicky, A. I. (2002). Stochastic natural language generation for spoken dialog systems. *Computer Speech & Language*, *16*(3), 387-407.

[17] Mellish, Chris, and Robert Dale. "Evaluation in the context of natural language generation." Computer Speech & Language 12.4 (1998): 349-373.

[18] Chambers, Nathanael, and James Allen. Stochastic language generation in a dialogue system: Toward a domain independent generator. FLORIDA INSTITUTE FOR HUMAN AND MACHINE COGNITION INC PENSACOLA FL, 2004.

[19] Papineni, K., et al. "Bleu: a method for automatic evaluation of machine translation, 2001." Available from: citeseer. ist. psu. edu/papineni02bleu. html(2001).

[20] Chen, Stanley F., Douglas Beeferman, and Roni Rosenfield. "Evaluation metrics for language models." (1998).

[21] Justo, R., & Torres, M. I. (2009). Phrase classes in two-level language models for ASR. *Pattern Analysis and Applications*, *12*(4), 427-437.

[22] Torres, I., & Varona, A. (2001). k-TSS language models in speech recognition systems. Computer Speech & Language, 15(2), 127-149.] for the Automatic Speech Recognition tasks.

[23] Torres, M. Inés, and Dpto Electricidad y Electrónica. "Stochastic Bi-Languages to model Dialogs." Finite State Methods and Natural Language Processing (2013): 9.

[24] Garcia, P., & Vidal, E. (1990). Inference of k-testable languages in the strict sense and application to syntactic pattern recognition. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 12(9), 920-925.

[25] Justo, R., & Torres, M. I. (2013). Hierarchical Models for Rescoring Graphs vs. Full Integration. In Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications (pp. 496-503). Springer Berlin Heidelberg.

[26] Benedı, José-Miguel, et al. "Design and acquisition of a telephone spontaneous speech dialogue corpus in Spanish: DIHANA." Fifth International Conference on Language Resources and Evaluation (LREC). 2006.

[27] Alcácer, N., Benedı, J. M., Blat, F., Granell, R., Martınez, C. D., & Torres, F. (2005). Acquisition and labelling of a spontaneous speech dialogue corpus. InProceeding of 10th International Conference on Speech and Computer (SPECOM). Patras, Greece (pp. 583-586).

[28] Cantone, D., Cristofaro, S., Faro, S., & Giaquinta, E. (2009, March). Finite State Models for the Generation of Large Corpora of Natural Language Texts. In Finite-state Methods and Natural Language Processing: Post-proceedings of the 7th International Workshop FSMNLP; Edited by Jakub Piskorski, Bruce Watson and Anssi Yli-Jyrä (Vol. 191, p. 175). IOS Press.

[29] Bangalore, Srinivas, Owen Rambow, and Steve Whittaker. "Evaluation metrics for generation." Proceedings of the first international conference on Natural language generation-Volume 14. Association for Computational Linguistics, 2000.