Language Models for Name Recognition in Spanish Spoken Dialogue Systems

German Tapia, Ivan V. Meza, and Luis Pineda

Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS),
Universidad Nacional Autónoma de México,
Circuito Escolar s/n, Ciudad Universitaria, D.F., Mexico
tapia_g@uxmcc2.iimas.unam.mx,ivanvladmir@turing.iimas.unam.mx,
luis@leibniz.iimas.unam.mx
http://www.iimas.unam.mx/

Abstract. Current advances on dialogue system require the development of language models for automatic speech recognition that are not only domain or task specific but also sub-task specific (e.g. name, age or price recognition). This paper presents a method for the creation of language models for name recognition at the greeting stage of a conversation in spoken Spanish. In particular, we focus on the introductory phrases (e.g., *mi nombre es John/my name is John*). The method relies in the construction of a corpus for names, and we study two variants; in one names are uniformly distributed; in the other, names are represented in a manner consistent with their frequency in web-searches.

Keywords: Automatic Speech Recognition, Language Model, Spoken Dialogue System.

1 Introduction

Current practice permits the development of dialogues system (i.e., oriented to a specific task) with common underlying principles in a reasonable amount time [10, 4, 7]. However, Automatic Speech Recognition (ASR) technology is still a bottleneck for this kind of systems [14, 7]. There are several possible actions to improve the ASR module of a spoken dialogue system (e.g., to improve the acoustic model, to create better dictionaries, to use n-best list, etc.) In this paper, we focus on improving the language models. In particular, on the creation of language models for a specific conversational domain.

A large amount of research has been focused on the automatic creation of domain specific languages models [2, 3, 6, 15]. However, these models cannot be generalized to other domains, and much work and effort is required for developing new applications. To overcome this problem, we explore the automatic creation of language models which are sub-task specific and can be used in different applications and domains. A sub-task in a dialogue system is a small interaction between the user and the system which aims to resolve a specific conversational goal. Examples of subtasks are asking for the user's name or age, or providing

© G. Sidorov et al. (Eds.)
Advances in Soft Computing Algorithms
Research in Computing Science 49, 2010, pp. 145-154

the price of a product. In particular, we focus on the name recognition sub-task which happens during the introductory stage of a conversation. In this sub-task the user says phrases like *I'm called.../me llamo...* when his or her names is asked for. This is an important sub-task in a conversation, since the name can be used productively later on in the conversation for several purposes. However, it is hard to identify a good language model for such task, since there are a great amount of possible names, which have a unusual frequency distribution, with a few popular names, but the rest are not so common.

There are two general approaches for automatic language generation: using hand-coded grammars and an using out-of-domain corpus. In the former case, a set of rules and an in-domain lexicon are used to generate an in-domain corpus. This approach has the disadvantage that the quality of the corpus depends on the skill of the grammar's designer. Additionally, it is hard to make sure that the corpus frequencies correspond to the language used in common conversations. In the latter case, a large out-of-domain corpus is used to enrich a small in-domain corpus. This small in-domain corpus is usually called the seed. The result is a subset of the out-of-domain corpus but much larger than the seed. In this work, we propose to build a set of rules for the generation of name recognition expressions as hand-coded grammars approaches do. However, we implement two versions of this approach. The first version uses a uniform distribution of names, while the second is parametrized by the proportions of names in an out-of-domain corpus.

This paper is organized as follows. Section 2 reviews previous work. Section 3 presents our general methodology and its application to the name recognition sub-task. Section 4 describes the corpora used during the experiments. Section 5 describes the experiments performed and the results obtained. Finally, section 6 presents our conclusion and further work.

2 Previous Work

Previous work has focused on the creation of domain specific language models. Galescu et al. proposed two methods. The first uses a context free grammars to generate sentences in the domain. The second uses an out-of-domain corpus to identify the sentences which fit the domain best [3]. However, they results show that the interpolation of language models resulting of the two methods provides the best performance. Chung et al. presents a technique where an out-of-domain corpus is transformed to fit the domain [2]. This transformation is performed with two techniques. A set of hand-coded rules are used to rewrite the sentences, or automatic translation techniques are used to "translate" the sentences into the domain. Mahajan et al. uses information retrieval techniques to find an in-domain examples from an out-of-domain corpus [6].

These efforts focus on domain specific language models. That means that they aim to capture language specific to a domain, so that if a dialogue system is about tickets reservation the resulting language model using these techniques will contain information about tickets reservation. However, for a different dialogue system in another domain it will be necessary to create a different language model. In this work we look at language models which can be shared among dialogue systems; these will cover small parts of the dialogue but can be used in other systems and domains. These models could be either loaded dynamically for the specific part of the conversation or can be interpolated with other models to compose a final language model specific to the dialogue system.

A similar problem appears in directory assistance [1] and local business voice search [14] tasks. In these, a user asks to a spoken system the location of a business using its name. Business names also have an unusual frequency distribution, with a few words occurring very often and the rest, rarely. For both tasks, it has been identified a low performance for the ASR module and it has been proposed to use a set of approaches relying in the creation of language models. In this paper, we present a method for building language models too, but in our case we focus on name recognition

3 Sub-task Specific Language Model

Figure 1 shows a typical architecture of a dialogue system. There are three main stages: language understanding, dialogue management and language generation. The first stage assigns an interpretation or meaning to the input message. The purpose of the second is to choose the action to be taken next. Finally, the language generation stage renders a meaningful message directed to the user. The dialogue management component rests on a conversational protocol that models the task structure (e.g., [8]). This is, the conversation is divided into sub-tasks. For instance the ATIS systems [13], divides the reservation task into flight, car and hotel reservation. These tasks can be further divided into sub-tasks that identify or resolve a relevant piece of information, For instance, the identification of the destination city for the flight reservation task.

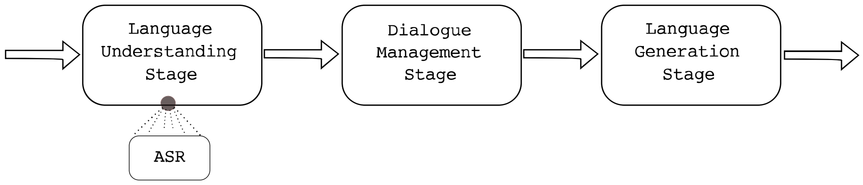
Language Understanding Stage
Dialogue Management Stage
Language Generation Stage
ASR

Fig. 1. Basic dialogue system architecture.

There are sub-tasks common to different systems. For instance, an introductory stage in which the user and the system introduce themselves to each other. This has as a further sub-task through which the user gives his or her name to the system. We called this sub-task *name recognition.* An instance of this is shown in Figure 2. The sub-task occurs between utterances 2 and 3. The name

No.	Speaker	Utterance
1	SYS	Hello, my name is Golem and this is the game Guess the card
2	SYS	There is a player. What is your name?
3	USR	Alba
4	SYS	Hello Alba, nice to meet you
5	SYS	How are you ?
...	...	...

Fig. 2. Excerpt of a introductory stage between a dialogue system (SYS) and user (USR).

recognition is important because the system can use the name to address the user later on, as in 4.

We have identified the name recognition sub-task in different dialogue systems: the Golem-2 system, a mobile robot which guides a poster session, asks for the user's name of the user [4]; the "Guess the card" system, which plays a game with children at a science museum also asks for the user's name [7]; the excerpt of the conversation in Figure 2 was taken from an actual conversation with this system. This sub-task occurs also in the *who-is-who* competition at *Robocup@home*[1] too. Here, a service robot looks for known and unknown participants within a scenario simulating a house. Once a participant is found the robot has to determine if the participant is known. If this is not the case, the robot has to introduce itself and ask for the participant's name, otherwise it has to address him or her by his or her name.

To address the problem of generation of sub-task specific language models we propose the mechanism showed on Algorithm 1. First a set of topics are identified. In the case of name recognition, there is only one topic involved: the *NAME*. For the step 2 a set of words belonging to the topic have to be defined. For the case of the *NAME* topic it is necessary to specify a list of names which will be identified. Additionally, we have to specify a set of templates on which such topics appear, like *my name is NAME*. In this template the *NAME* tag will be replaced by a name from the list. The next stage is to collect a set of statistics of the words conditioned on the topic. In this paper, we propose two strategies; the first consists of using an *uniform* distribution where all names appear with the same frequency. The second consists on using *web-searches hits* to approximate the distribution of the names. Once the corpus is available, a language model is generated using standard techniques. In particular, we explore the impact of using bigrams or trigrams for the language model.

[1] http://www.ai.rug.nl/robocupathome/

Algorithm 1 Language model generation.

```
1: Create a set of goal topics.
2: For each topic create a list of words belonging to such topic.
3: Create a set of templates which use such topics.
4: Obtain statistics of the words conditioned on the topic from an out-of-domain
   resource.
5: Generate sentences for the language model by substituting the words on the tem-
   plates and by using the statistics of the words.
6: Generate a language model with the sentences generated in the previous words
   (using bigrams or trigrams).
```

4 Corpora

The algorithm 1 is used to generate a corpus until the step 5[2]. This corpus is then used in step 6 to generate a language model. We look into different strategies to generate this corpus. As mentioned two strategies are used to generate three corpora. All of these corpora uses the Spanish templates: *me llamo NAME/I'm called NAME*, *mi nombre es NAME/my name is NAME* and *NAME*. Recall that the*NAME* tag in the template will be replace by a name. We used a list of 624 names which were already available as a part of a functional dialogue system to generate $1,872$ sentences [7]. We also used *web-searches* hits of these 624 names to create a corpus. In this case, we repeated the phrases at least three times to match the distribution of names with the one of the web-searches. Table 1 shows the main properties of the three corpora. The first corpus uses the uniform strategy and we called it *small uniform*. However, this version of the corpora is small when compared with the third which uses web-searches hits. For these reason, we generated a *large uniform* by repeating each template $1,000$ times.

Table 1. Characterization of generated corpora.

	Small uniform	Large uniform	Web-searches
Word types	624	624	624
Sentences	1,872	1,872,000	2,875,884
Bigrams	2,474	2,474	2,474
Trigrams	4,324	4,324	4,324

We have also collected two audio corpora for testing the language models. The first consists of a collection of recordings of *final-users* interactions with a functioning dialogue system [7]. In particular, we identify the utterances in which the users utter their names. These audios have been manually transcribed with their orthographic transcription. However, this is a small corpus since we

[2] An implementation of this algorithm and the scripts used during the experiments can be found in the following link `http://code.google.com/p/language-models/downloads/`.

do not have that many instances of people saying their name to the system. On the other hand, we created an *artificial* generated corpus in which we ask ten users to tell ten unique random names in three different forms. With this setting we collected 300 instances of people uttering a name. Table 2 summarises the main propertied of these two corpora.

Table 2. Characterization of artificial and final-user corpora.

	Artificial corpus	Final-users corpus
Total participants	10	12
Total recordings	300	16
Total ways of uttering names	3	1
Total different names	100	12

Additionally, we use the DIMEx100 corpus to build the acoustic models used in the experiments [12]. The DIMEx100 corpus is a collection of 100 persons reading 50 different sentences and 10 common sentences in Spanish, phonetically transcribed. The corpus is phonetically balanced and has been used for the development of the Spanish ASR systems for the "Guess the card" and Golem systems. A speech recogniser developed with this corpus can be expected to have a performance of at least 48.3 Word Error Rate (WER) using a weak language model.

5 Experiments and Results

We generated six sub-task specific language models for name recognition using the three corpora described in section 4, with bigrams and trigrams. For testing we used the Sphinx 3 ASR system [5]. This setting allows us to measure the direct effect of the language models on the performance of the ASR module.

The first experiment consisted on measuring the WER of the six language models. We used the *artificial* generated corpus for testing. Additionally, we varied the language weight on the ASR module. This weight is used during decoding to couple the acoustic models with the language models [3]. The results are shown on Table 3. These results show better performance than previously reported [9]. However, it is important to keep in mind that the sentences on the *artificial* corpus are simpler. We also observe that the greater the weight the better the performance of the uniform models. This is not the case for the *web-searches* models which reach a high point at the weight of 7. However, we found that large language models, *large uniform* and *web-searches*, perform better small ones.

The WER gives a good measure of the ASR task; however it does not tell us much about the name recognition task. For this, we measured the performance

[3] We vary the weight from 5 to 13 which is the recommended range for Sphinx 3.

Table 3. WER for language models using *artificial* generated names corpus (the lower the better).

Language weight	Small Uniform		Large Uniform		Web-searches	
	Bigram	Trigram	Bigram	Trigram	Bigram	Trigram
5	35.1	35.9	26.0	26.6	26.1	27.1
7	29.8	30.7	25.6	24.9	**25.5**	**25.7**
9	28.2	28.7	25.3	25.4	25.6	25.7
11	27.1	27.4	25.6	25.4	25.8	25.8
13	**26.2**	**26.2**	**25.1**	**25.1**	26.4	26.4

of the language models on their capacity of recovering the right name and identifying the right context of the name words, this is the words which are or not a name. For this we defined two measures:*Name recovery* (*NR*) and *Name and context recovery* (*NCR*). *Name recovery* quantifies how many times the name was correctly retrieved from the hypothesis of the ASR system. *Name and context recovery*[4] quantifies how many times the right name and the context words were recovered from the recognition hypothesis; in this case we can calculate a precision, recall and $F1$-score. Table 4 presents the *name recovery* score and Table 5 the $F1 - score$ for *name and context recovery.* Similarly, as in the previous experiments, we show the results for different language weights.

Table 3 shows that the ASR performance was good. However, the *Name recovery* scores in Table 4 show that name recognition performance should be improved. The *uniform* language models outperform the *web-searches* ones. On the other hand, 5 shows that the *large* language models outperform the *small* ones; this measurement is closer to WER results showed in Table 3. We think that this change of better performance from uniform to large language models is due to the nature of the measurement. The WER and *Name and Context Recovery* scores are global measures. These consider the name recognition part, but they also consider the no-name words. On the other hand, *Name recovery* only considers the names. A large model will have better statistics about the context words than a small one. A uniform model will give any name a better chance to occur. This is consistent with the *large uniform* language model's good performance on both measurements. This is because it combines two important properties: uniformity and size.

In the second experiment, we evaluated the performance of the six language models using the *final-users* corpus. Table 6 presents a summary of the results. We use the better scores for the language weight and n-gram as shown in Table 3; we we used bigrams in all cases. Here, a pattern compatible with *artificial* corpus can be seen. They perform similarly for the WER score; however the *large uniform* corpus gives the best performance. For the name recovery score the

[4] Notice, that although we focus on the name recognition part, it is important to consider how good the approach is for the rest of the expressions, since they also could contain useful information.

Table 4. Name recovery scores for language models using *artificial* generated names corpus (the larger the better).

Language weight	Small Uniform		Large Uniform		Web-searches	
	Bigram	Trigram	Bigram	Trigram	Bigram	Trigram
5	33.6	32.3	34.0	34.0	**33.0**	**33.0**
7	34.0	33.6	**34.3**	**34.3**	32.6	32.6
9	34.0	34.0	34.0	34.0	30.3	30.3
11	34.3	34.3	33.3	33.3	29.3	29.3
13	**34.3**	**34.3**	32.6	32.6	27.6	27.6

Table 5. Name and context recovery $F1$-scores for language models using *artificial* generated names corpus (the larger the better).

Language weight	Small Uniform		Large Uniform		Web-searches	
	Bigram	Trigram	Bigram	Trigram	Bigram	Trigram
5	58.7	57.8	71.11	70.83	71.0	70.1
7	64.1	64.0	72.61	72.34	**71.5**	**71.4**
9	66.4	66.2	71.93	72.07	71.0	71.0
11	68.4	68.3	**72.34**	**72.34**	70.7	70.7
13	**69.6**	**69.6**	71.93	71.93	69.9	69.9

small uniform language perform better than the others. However, for the more global score, NCR, the situation is reversed and the larger corpora outperforms the smaller.

Table 6. Scores for language models using *final-user* corpus (WER, the smaller the better; NR and NCR, the larger the better).

Language models	WER	NR	NCR
Small uniform	39.6	**31.3**	43.9
Large uniform	**37.5**	25.0	**51.2**
Web-searches	39.6	25.0	**51.2**

6 Conclusion and Future Work

In this paper we presented a method to generate sub-task specific dialogue models. The difference of this work with previous approaches is that it builds small language models which can be shared among different dialogue systems which have the same sub-task. In particular, we focus on the name recognition sub-task. This sub-task is part of an introductory stage in conversation. In this sub-task the user gives his or her name to the system. It is important to identify the name since the system can address the user by his or her name later on during the conversation. This sub-task is common among different dialogue systems in which to know the name of the user is required. On the other hand, names are challenging to model since there is a large amount of names and their frequency distribution is unusual, with few names with high frequency and the rest with extremely low frequencies.

We explored two strategies to generate three corpora for the language models. The first consisted in using a uniform distribution for the names. The second consisted in using an approximate distribution for the names based on web-searches hits. We generated three corpora, a *small uniform*, a *large uniform* and *web-searches*. With these three corpora we generated six language model, using bigrams and trigrams for each of the three corpora. These six language models allows us to measure the effect of the strategy on the ASR module.

For testing two corpora were collected. The first is an *artificial* corpus in which 10 subject utter 10 names in three different ways. The second is a *final-user* corpus in which an actual users utter their name to a dialogue system. We tested the ASR performance using the different language models in three different ways. We first used the standard Word Error Rate (WER). The second consisted on identifying how many names could be recovered from the ASR hypothesis (i.e., Name recovery, *NR*). Finally, the third measurement consisted on comparing the times the names could be recovered and the times the context words could be recovered (i.e., Name and context recovery, *NCR*). The main difference between the two last measurements is that the first is local to the recognition of the names, and second is global to the task of recovering information.

Our experiments showed that the combination of a large corpus using an uniform distribution provided the best balance for the language model. In particular, we obtained good local (*NR*) and global scores (WER and *NCR*). The *web-searches* corpora was not good to capture a good performance when trying a local score as the *NR*. Another important result was to discard the trigrams for futures implementations of the language models, since the best performance is reachable using the less complex bigrams. This is consistent with our intuition about the name recognition sub-task on which we rarely find more than four words for the involved phrases (e.g., *mi nombre es NAME*/my name is).

We have also identified points of further improvement and new research questions. We explored two strategies to generate the corpora for the language models so far. However, we could get a better approximation of the names by using other source of information, for instance name statistics from schools or the national registry. We are also looking into complement our *final-users* corpus with a larger amount of examples of people uttering their names in functioning dialogue system. We are also looking into generating language models for other sub-tasks like age identification. We also need to test these language models in a working dialogue system. We plan to test the dynamic option, in which a specific language model is loaded for a particular part of the conversation, and the interpolation option, in which a set of language models are interpolated to create a language model specific to a dialogue system.

Acknowledgements. We thank the enthusiastic support of the members of the DIME and Golem group at IIMAS and at Science Museum Universum-UNAM. We also gratefully thank the support of grants CONACYT 81965 and PAPPIT-UNAM IN-104408 and IN-115710.

References

1. Buntschuh, B., Kamm, C., Di Fabbrizio, G., Abella, A., Mohri, M., Narayanan, S., Zeljkovic, I., Sharp, R., Wrigth, J., Marcus, S., Shaffer, J., Duncan, R., Wilpon, J.: VPQ: a spoken language interface to large scale directory information. In: Proceedings of ICSLP (1998)
2. Chung, G., Seneff, S., Wang., C.: Automatic Induction of Language Model Data for a Spoken Dialogue System. In: Proc. SIGdial-05 (2005)
3. Galescu, L., Ringger, E., Allen, J.: Rapid Language Model Development for New Task Domains. In: Proc. LREC, Granada (1998)
4. Avilés, H., Alvarado-González, M., Venegas, E., Rascón, C., Meza, I., Pineda, L. A.: Development of a Tour–Guide Robot Using Dialogue Models and a Cognitive Architecture. In: Proceedings of Iberamia 2010, LNAI (2010)
5. Huerta, J. M., Chen, S.J., Stern R. M.: The 1998 carnegie mellon university sphinx-3 spanish broadcast news transcription system. In: Proc. of the DARPA Broadcast News Transcription and Understanding Workshop (1999)
6. Mahajan, M., Beeferman, D., Huang, D.: Improved Topic-Dependent Language Modeling Using Information Retrieval Techniques. In: Proc. ICASSP, pp. 541-544 (1999)
7. Meza, I., Salinas, L., Venegas, E., Castellanos4, H., Chavarría, A., Pineda L. A.: Specification and Evaluation of a Spanish Conversational System Using Dialogue Models. In: Proceedings of Iberamia 2010 (2010)
8. Pineda, L. A.: Specification and Interpretation of Multimodal Dialogue Models for Human-Robot Interaction. In: G. Sidorov (Ed.) Artificial Intelligence for Humans: Service Robots and Social Modeling, SMIA, México, pp. 3350 (2008)
9. Pineda, L. A., Castellanos, H., Cuétara, J., Galescu, L., Juárez, J., Llisterri, J., Pérez-Pavón, P., Villaseñor, L.: The Corpus DIMEx100: Transcription and Evaluation. Language Resources and Evaluation (2009)
10. Pineda, L. A., Meza, I., Salinas, L.: Dialogue Model Specification and Interpretation for Intelligent Multimodal HCI. In: Proceedings of Iberamia 2010, LNAI (2010)
11. Oparing, I.: Statistical Language Models for Dialogue Systems. Technical Report (2006)
12. Pineda, L. A., Castellanos, H., Cuétara, J., Galescu, L., Juárez, J., Llisterri, J., Pérez-Pavón, P., Villaseñor, L.: The Corpus DIMEx100: Transcription and Evaluation. Language Resources and Evaluation (2009)
13. Deborah A. Dahl et al.: ATIS 3 training data. Linguistic Data Consortium, Philadelphia (1994)
14. Stent, A., Zeljković, I., Caseiro, D., Wilpon, J.: Geo-Centric Language Models for Local Business Voice Search. In: Proceedings of HLT/NAACL-2009 (2009)
15. Wan, V., Hain, T.: Strategies for language model web-data collection. In: Proc. ICASSP, UK (2006)