Semi-Supervised Approach to Named Entity Recognition in Spanish Applied to a Real-World Conversational System

Víctor R. Martínez[1(✉)], Luis Eduardo Pérez[1], Francisco Iacobelli[2], Salvador Suárez Bojórquez[3], and Víctor M. González[1]

[1] Department of Computer Science, Instituto Tecnologico Autonomo de Mexico, Rio Hondo #1, Progreso Tizapan, Del. Alvaro Obregon, 01080 Mexico City, Mexico
{victor.martinez,luis.perez.estrada,victor.gonzalez}@itam.mx
[2] Northwestern University, Frances Searle Building 2-343, 2240 Campus Drive, Evanston, IL, USA
f-iacobelli@u.northwestern.edu
[3] BlueMessaging, Ibsen 40, Segundo Piso Col. Polanco, Del. Miguel, Mexico City, Mexico
info@bluemessaging

Abstract. In this paper, we improve the named-entity recognition (NER) capabilities for an already existing text-based dialog system (TDS) in Spanish. Our solution is twofold: first, we developed a hidden Markov model part-of-speech (POS) tagger trained with the frequencies from over 120-million words; second, we obtained 2,283 real-world conversations from the interactions between users and a TDS. All interactions occurred through a natural-language text-based chat interface. The TDS was designed to help users decide which product from a well-defined catalog best suited their needs. The conversations were manually tagged using the classical Penn Treebank tag set, with the addition of an ENTITY tag for all words relating to a brand or product. The proposed system uses an hybrid approach to NER: first it looks up each word in a previously defined catalog. If the word is not found, then it uses the tagger to tag it with its appropriate POS tag. When tested on an independent conversation set, our solution presented a higher accuracy and higher recall rates compared to a current development from the industry.

1 Introduction

Text-based Dialog Systems (TDS) help people accomplish a task using written language [24]. A TDS can provide services and information automatically. For example, many systems have been developed to provide information and manage flight ticket booking [22] or train tickets [11]; decreasing the number of automobile ads according to the user preferences [9], or inquiring about the weather

Víctor R. Martínez and Luis Eduardo Pérez—These authors contributed equally to the work.

J.A. Carrasco-Ochoa et al. (Eds.): MCPR 2015, LNCS 9116, pp. 224–235, 2015.
DOI: 10.1007/978-3-319-19264-2_22

report in a certain area [25]. Normally, these kinds of systems are composed by a dialog manager, a component that manages the state of the dialog, and a dialog strategy [12]. In order to provide a satisfactory user experience, the dialog manager is responsible for continuing the conversation appropriately, that is reacting to the user's requests and staying within the subject [16]. To ensure this, the system must be capable of resolving ambiguities in the dialog, a problem often referred to as Named Entity Recognition (NER) [10].

Named Entity Recognition consists in detecting the most salient and informative elements in a text such as names, locations and numbers [14]. The term was first coined for the Sixth Message Understanding Conference (MUC-6) [10], since then most work has been done for the English language (for a survey please refer to [17]). Other well represented languages include German, Spanish and Dutch [17].

Feasible solutions to NER either employ lexicon based approaches or multiple machine learning techniques, with a wide range of success [14,17]. Some of the surveyed algorithms work with Maximum Entropy classifiers [3], AdaBoost [4], Hidden Markov Models [1], and Memory-based Learning [21]. Most of the novel approaches employ deep learning models to improve on accuracy and speed compared to previous taggers [5]. Unfortunately, all these models rely on access to large quantities (in the scale of billions) of labeled examples, which normally are difficult to acquire, specially for the Spanish language.

Furthermore, when considering real-world conversations, perfect grammar cannot be expected from the human-participant. Errors such as missing letters, lacking punctuation marks, or wrongly spelled entity names could easily cripple any catalog-based approach to NER. Several algorithms could be employed to circumvent this problems, for example, one could spell-check and automatically replace every error or consider every word that is within certain distance from the correct entity as an negligible error. However, this solutions do not cover the whole range of possible human errors, and can be quite complicated to maintain as the catalog increases in size.

In this work we present a semi-supervised model implementation for tagging entities in natural language conversation. By aggregating the information obtained from a POS-tagger with that obtained from a catalog lookup, we ensure the best accuracy and recall from both approaches. Our solution works as follows: first, we look each word coming from the user in our catalog of entities. If the word is not present, we then use a POS-tagger to find the best possible tag for each word in the user's sentence. We use the POS tag set defined by the Penn Treebank [15] with the addition of an ENTITY tag. We trained our POS-tagger with 120-million tagged words from the Spanish wiki-corpus [20] and thousands of real-world conversations, which we manually tagged.

Our approach solves a specific problem in a rather specific situation. We aimed to develop a tool that is both easy to deploy and easy to understand. In a near future, this model would provide NLP-novice programmers and companies with a module that could be quickly incorporated as an agile enhancement for the development of conversational agents focused on providing information

on products or services with a well-defined catalog. Examples of this scenario include the list of movies shown in a theater, or different brands, models and contracts offered by a mobile phone company. To the best of the authors knowledge, no similar solution has been proposed for the Spanish language. Further work could also incorporate our methodology into a Maximum Entropy Classifier, an AdaBoost, or even a DeepLearning technique to further improve on the entity detection.

The paper is structured as follows: in Sect. 2 we present the POS tagging problem and a brief background of POS taggers using hidden Markov models. In Sect. 3 we present the obtained corpus and data used along this work. Section 4 presents the methodology for this work, while Sect. 5 shows our main results. Finally, Sect. 6 describes the real-world implementation and compares our work against the industry's current development.

2 Formal Background

2.1 Part of Speech Tagger

In any spoken language, words can be grouped into equivalence classes called parts of speech (POS) [12]. In English and Spanish some examples of POS are *noun*, *verb* and *articles*. Part-of-speech tagging is the process of marking up a word in a text as corresponding to a particular part of speech, based on both its definition, as well as its context (*i.e.*, its relationship with adjacent and related words in a phrase or sentence) [12]. A POS tagger is a program that takes as input the set of all possible tags in the language (*e.g.*, noun, verbs, adverbs, etc.) and a sentence. Its output is a single best tag for each word [12]. As aforementioned, ambiguity makes this problem non trivial, hence POS-tagging is a problem of disambiguation.

The first stage of our solution follows the classical solution of using a HMM trained in a previously tagged corpus in order to find the best relation between words and tags. Following are the formal definitions used in our work.

Hidden Markov Models. Loosely speaking, a Hidden Markov Model (HMM) is a Markov chain observed in noise [19]. A discussion on Markov chains is beyond the scope of this work, but the reader is referred to [13,18] for further information. The underlying Markov chain, denoted by $\{X_k\}_{k\geq 0}$ is assumed to take values in a finite set. As the name states, it is *hidden*, that is, it is not observable. What is available to the observer is another stochastic process $\{Y_k\}_{k\geq 0}$ linked to the Markov chain in that X_k governs the distribution of Y_k [19]. A simple definition is given by Capp [19] as follows:

Definition 1. *A hidden Markov Model is a bivariate discrete time process* $\{X_k, Y_k\}_{k\geq 0}$ *where* $\{X_k\}$ *is a Markov chain and, conditional on* $\{X_k\}$, $\{Y_k\}$ *is a sequence of independent random variables. The state space of* $\{X_k\}$ *is denoted by* X*, while the set in which* $\{Y_k\}$ *takes its values is denoted by* Y*.*

We use this definition as it works for our purpose and has the additional benefit of not overwhelming the reader with overly complex mathematical terms. Following this definition, the problem of assigning a sequence of tags to a sequence of words can be formulated as:

Definition 2. *Let Y be the set of all possible POS tags in the language, and $w_1, w_2, w_3, \ldots, w_n$ a sequence of words. The POS-tagging problem can be expressed as the task of finding for each word w_i, the best tag $\hat{y}_i$ from all the possible tags $y_i \in Y$. That is,*

$$\hat{y}_i = \underset{y_i \in Y}{\operatorname{argmax}} P(y_i \mid w_i) \tag{1}$$

Given a sequence of words $w_1, w_2, \ldots, w_T$, an equivalent approach to the POS tagging problem is finding the sequence $y_1, y_2, \ldots, y_T$ such that the joint probability is maximized. That is,

$$\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_T = \underset{y_1, y_2, \ldots, y_T}{\operatorname{argmax}} P(w_1, w_2, \ldots, w_T, y_1, y_2, \ldots, y_T) \tag{2}$$

We then can assume that the joint probability function 2 is of the form

$$P(w_1, w_2, \ldots, w_T, y_1, y_2, \ldots, y_T) = \prod_{i=1}^{T+1} q(y_i \mid y_{i-2}, y_{i-1}) \prod_{i=1}^{T} e(w_i \mid y_i) \tag{3}$$

with $y_0 = y_{-1} = *$ and $y_{T+1} = STOP$.

The function q determines the probability of observing the tag y_i after having observed the tags y_{i-1} and y_{i-2}. The function e returns the probability that the word w_i has been assigned the label y_i. By maximizing both functions, the joint probability P is maximized. Now we explain how we can obtain both 4 and 5 from a previously tagged corpus.

Equation 3 captures the concept of using word-level trigrams for tagging [7]. Calculating the function q will depend on the two previous tags. Using the definition of conditional probability, we observe that the function q is the ratio between the number of times y_i is observed after y_{i-2}, y_{i-1} and how many times the bigram y_{i-2}, y_{i-1} was observed in the corpus:

$$q(y_i \mid y_{i-2}, y_{i-1}) = \frac{q(y_i, y_{i-2}, y_{i-1})}{q(y_{i-2}, y_{i-1})} \tag{4}$$

Similarly, the probability that w_i is labelled with y_i can be obtained as the frequentist probability observed in the corpus

$$e(x_i \mid y_i) = \frac{\#\text{ times } x_i \text{ was seen tagged as } y_i}{\#\text{ times} x_i \text{ occurs in the corpus}} \tag{5}$$

Viterbi Algorithm. The Viterbi algorithm, proposed in 1966 and published the following year by Andrew J. Viterbi, is a dynamic programming algorithm designed to find the most likely sequence of states, called the Viterbi path, that could produce an observed output [23]. It starts with the result of the process (the sequence of outputs) and conducts a search in reverse, in each step discarding every hypothesis that could not have resulted in the outcome.

The algorithm was originally intended for message coding in electronic signals, ensuring that the message will not be lost if the signal is corrupted, by adding redundancy. This is called an error correcting coding. Today, the algorithm is used in a wide variety of areas and situations [8]. Formally, the Viterbi algorithm is defined as follows

Definition 3. *Let $\lambda = \{X_k, Y_k\}_{k \geq 0}$ be an HMM with a sequence of observed inputs $y_1, y_2, \ldots, y_T$. The sequence of states $x_1, x_2, \ldots, x_T$ that produced the outputs can be found using the following recurrence relation:*

$$V_{1,k} = P(y_1 \mid k) \cdot \pi_k \tag{6}$$

$$V_{t,k} = P(y_t \mid k) \cdot \max_{s \in S}(A_{s,k} \cdot V_{t-1,s}) \tag{7}$$

where $V_{t,k}$ is the probability of the most likely sequence of states that result in the first t observations and has k as a final state, and $A_{s,k}$ is the transition matrix for $\{X_k\}$

The Viterbi path can be found following backward pointers, which keep a reference to each state s used in the second equation. Let $Ptr(k, t)$ be a function that returns the value of s used in the calculation of $V_{t,k}$ if $t > 1$, or k if $t = 1$. Then:

$$s_T = \operatorname*{argmax}_{s \in S}(V_{T,s}) \tag{8}$$

$$s_{t-1} = Ptr(s_t, t) \tag{9}$$

3 Data

We propose an implementation of the Viterbi algorithm to tag every word in a conversation with a real-world dialog system. The resulting HMM corresponds to an observed output $w_1, w_2, \ldots, w_T$ (the words in a conversation), and the states $y_1, y_2, \ldots, y_T$ (the POS-tags for each word). Some simplifications and assumptions allowed us to implement the Markov model starting from a set of data associating each word (or tuple of words) with one (or more) tags, often called a tagged corpus.

For this work, we used the Spanish Wikicorpus [20], a database of more than 120 million words obtained from Spanish Wikipedia articles, annotated with lemma and part of speech information using the open source library FreeLing. Also, they have been sense annotated with the Word Sense Disambiguation algorithm UKB. The tags used and the quantity of corresponding words in the corpus are given in Table 1. We can note that the least common tag in this corpus is dates (**W**), with no examples in the texts used. We suspect this was due an error on our side while parsing the corpus.

Table 1. Number of words per tag in WikiCorpus [20]

tag	POS	count
A	Adjective	4,688,077
R	Adverb	1,901,136
D	Determinant	10,863,584
N	Noun	21,651,297
V	Verb	8,445,600
P	Pronoun	2,832,306
C	Conjunction	3,413,668
I	Interjection	33,803
S	Preposition	11,773,140
F	Punctuation	5,734
Z	Numeral	1,828,042
W	Date	0

3.1 Rare Words

Even with a corpus of 120 million words, it is quite common that conversations with users contain words that never appeared in the training set. We looked into two possible ways of handling these words: replacing all words with their corresponding tokens, and replacing only *rare* words with their respective tokens. A word is considered *rare* if it's total frequency (number of times it appears in the corpus, regardless of its tag) is less than 5. For this work we only used 5 tokens (Table 2).

Table 2. Translation from words to placeholder tokens for handling words that were not observed in training.

Word	Translation (token)
Number with four digits	_4_DIGITS_
Any other number	_NUMBER_
Word contains a hyphen	_HYPHENATED_
Word is a Roman numeral	_ROMAN_
Word's total frequency < 5	_RARE_

4 Methodology

We downloaded the Spanish WikiCorpus in its entirety, transformed its files into a frequency counts file using a MapReduce scheme [6]. Our aggregated file

contains the number of times each word appears, and the number of times each word appears with a certain tag (*e.g.*, the number of times "light" appears as noun). We then applied the rules for rare words, as discussed in Sect. 3.1, which yielded two new corpus and frequency files upon which we trained our HMM.

Each of these two models was tested using five rounds of cross-validation over the whole WikiCorpus of 120 million words. For each round, our program read the frequency count file and calculated the most likely tag for each word in the training set. The result of each round was a confusion matrix. For each tag, we obtained measurements of the model's precision, recall, and F-score over that tag.

5 Results

5.1 Replacing only Rare Words with Tokens (VM1)

For words labeled as nouns or names (N), the algorithm had an average precision of 12.15 % and recall of 46.87 %. From the total of 21, 651, 297 names, it only managed to find 10, 147, 962. From the words the algorithm tagged as nouns or names, only 3 out of every 25 was, in fact, a noun or a name. The left side of Table 3 shows the results for this algorithm.

5.2 HMM Replacing Any Word in Table 2 with a Token (VM2)

This model stands out because of an increase in precision and recall in the classification of all tags. This comes as no surprise since the last rule in our translation Table 2 already considers the whole set of translated words in the previous model. The results are shown on the right side of Table 3.

6 Implementing in a Real-World Scenario

Considering the results described in the previous section, we decided to compare and improve model VM2 in a real world scenario. Such opportunity was presented by BlueMessaging Mexico's text-based dialog systems.

BlueMessaging is a private initiative whose sole objective is to connect business and brands with customers anytime anywhere [2]. It has developed several TDS systems, one of such (named CPG) was designed for helping users decide which product best fits their needs. CPG works with a well-defined product catalog, in a specific domain, and interacts with the user through a natural-language text-based chat interface. For example, CPG could help a user select among hundreds of cell phone options by guiding a conversation with the user about the product's details (see Table 4).

During the course of this work, we collected 2, 283 real-world conversations from CPG. This conversations were manually tagged over the same tag set as the wiki corpus. Words that were part of a product's name or brand were tagged as ENTITY. For example, *Samsung Galaxy S3* were labeled as *Samsung/ENTITY*

Table 3. Classification results for VM1 and VM2

	VM1			VM2		
	Precision	Recall	F-score	Precision	Recall	F-score
Adjective (A)	81.28 %	7.23 %	13.28 %	98.13 %	8.72 %	16.01 %
Conjunction (C)	4.87 %	23.85 %	8.08 %	20.64 %	97.71 %	34.04 %
Determinant (D)	7.74 %	32.38 %	12.48 %	23.18 %	98.53 %	37.49 %
Punctuation (F)	1.18 %	11.10 %	2.13 %	7.71 %	60.83 %	13.56 %
Interjection (I)	2.72 %	3.43 %	3.02 %	13.58 %	18.29 %	15.30 %
Noun (N)	12.15 %	46.87 %	19.27 %	23.40 %	95.68 %	37.56 %
Pronoun (P)	4.13 %	22.04 %	6.95 %	18.11 %	92.32 %	30.25 %
Preposition (S)	8.04 %	33.04 %	12.91 %	23.88 %	99.80 %	38.50 %
Adverb (R)	3.04 %	20.26 %	5.28 %	17.03 %	94.20 %	28.81 %
Date (W)	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %
Verb (V)	6.93 %	31.12 %	11.33 %	21.75 %	93.02 %	35.21 %
Numeral (Z)	5.89 %	27.78 %	9.70 %	23.45 %	99.04 %	37.88 %

Table 4. A translated conversation collected from BlueMessaging Mexico's CPG TDS

User	CPG
Hello	Welcome to CPG. Are you looking for a cell phone?
Do you have the iPhone?	Which iPhone are you looking for? 4s, 5s, 6?
6	Would you like the small or the big screen?

Galaxy/ENTITY S3/ENTITY. After this process, we had a new corpus containing 11,617 words (1,566 unique), from which 1,604 represent known entities.

We then tested three new approaches. The baseline was set by CPG's current catalog approach. We then tested our VM2 model, having scored the best across the tests in the last section. Finally, an hybrid approach was used: using both the POS-tagger and the catalog to identify named entities in the dialog. Each approach was tested with 5 rounds of cross-validation on the conversation corpus. Here we present the results for each of the three.

6.1 Catalog-Based Tag Replacement (DICT)

We tested a simple model consisting only of tag substitution using a previously known catalog of entities. Each entry in this dictionary is a word or phrase that identifies an entity, and an entity can be identified by multiple entries. This is useful for compound product names that could be abbreviated by the user. For example: "galaxy fame", "samsung fame" and "samsung galaxy fame" are all references to the same product. In this case, we used a catalog with 148 entries, identifying 45 unique entities.

To make the substitutions, the sentence to be labeled is compared against every entry in the catalog. If the sentence has an entry as a sub-sequence, the words in that sub-sequence are tagged as ENTITY, overriding any tags they might have had.

Using only this approach to tagging sentences, we reached a precision of 93.56 % and recall of 64.04 % for entities. However, this process does not give any information about any other tag, and does not attempt to label words as anything other than entity (Table 5).

Table 5. Classification results for DICT

	Precision	Recall	F1
ENTITY	93.56 %	64.04 %	76.03 %

6.2 Model VM2 Applied to Real-World Conversations (VM2-CPG)

Having tested this model with another corpus, we can use those results to compare its performance on the corpus of conversations, and see whether the model is too specific to the first corpus. Once again, we tested it using 5 rounds of cross-validation, training with subset of the corpus and leaving the rest as a testing set, for each round. The results for this model are shown on the left part of Table 6. With respect to the previous corpus, the model VM2 in this instance had less recall, but a better precision for nouns, determinants, interjections, pronouns, and adverbs.

6.3 VM3 Model with Catalog-Based Tag Replacement (VM2-DICT)

Lastly, we present the results for the model combining the HMM model, replacing words with the appropriate tokens, and tag replacement using a catalog of known entities as shown on the right hand of Table 6. This method is capable of identifying 24 out of every 25 entities.

7 Comparison with Current System

As a final validation for our models, we compared their performance against the system currently in use by BlueMessaging Mexico's platform. We collected an additional 2,280 actual user conversations. Again, we manually tagged the sentences according to the wikiCorpus tag set with the inclusion of the ENTITY tag. For both systems, we measured precision and recall. For the VM2-DICT model, the standard error was determined as two standard deviations from the results of the 5-fold cross-validation. Figure 1 shows the results obtained for each system. We found a highly significant difference in precision and recall of the two models (t-test $t = 23.0933$, $p < 0.0001$ and $t = 4.8509$, $p < 0.01$), with VM2-DICT having a better performance in both.

Table 6. Classification results for VM2 tested with the CPG corpus and for VM2 with dictionary replacement

	VM2 over CPG			VM2 with DICT		
	Precision	Recall	F1	Precision	Recall	F1
Adjective (A)	22.83 %	65.30 %	33.84 %	26.21 %	64.83 %	37.33 %
Conjunction (C)	76.22 %	66.75 %	71.17 %	75.68 %	68.19 %	71.74 %
Determinant (D)	63.37 %	43.81 %	51.80 %	64.22 %	45.32 %	53.14 %
Punctuation (F)	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %
Interjection (I)	88.79 %	78.37 %	83.25 %	89.77 %	77.61 %	83.08 %
Noun (N)	77.29 %	60.64 %	67.96 %	77.10 %	60.62 %	67.88 %
Pronoun (P)	84.53 %	52.33 %	64.64 %	84.59 %	53.68 %	65.68 %
Preposition (S)	56.12 %	62.35 %	59.07 %	58.07 %	62.06 %	60.00 %
Adverb (R)	73.40 %	53.86 %	62.13 %	75.63 %	52.45 %	61.94 %
Date (W)	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %
Verb (V)	84.48 %	54.13 %	65.98 %	84.06 %	53.87 %	65.66 %
Numeral (Z)	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %
Entity (ENTITY)	94.66 %	60.47 %	73.80 %	96.53 %	91.31 %	93.85 %

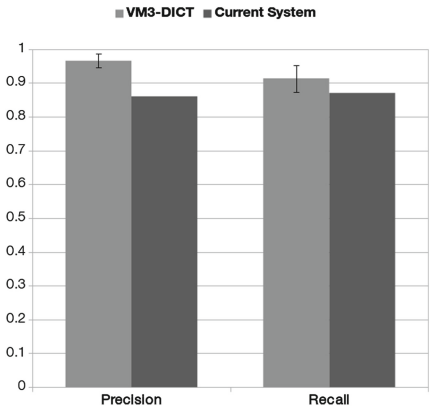

Fig. 1. Comparison between model VM2-dict and BlueMessaging Mexico's current system. The error bars show two standard deviations from the mean.

8 Discussion

We presented a semi-supervised approach to named entity recognition for the Spanish language. We noted that previous works on Spanish entity recognition used either a catalog-based approach or machine learning models with a wide range of success. Our model leverages on both approaches, improving both the accuracy and recall of a real-world implementation.

With a real-world implementation in mind, where solutions are measured by their tangible results and how easily they can be adapted to existing production schemes, we designed our system to be an assemblage of well-studied techniques requiring only minor modifications. We believe that our solution would allow for quick development and deployment of text-based dialog systems in Spanish. In a further work, this assemblage of simple techniques could evolve into more robust solutions, for example by exploring conditional random fields in order to replace some of the hidden Markov model assumptions. Moreover, implementations for deep learning in Spanish language might be possible in a future, as more researchers work in developing tagged corpus.

Acknowledgments. The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. We would also like to thank Consejo Nacional de Ciencia y Tecnología (CONACYT), BlueMessaging Mexico S.A.P.I. de C.V., and to the Asociación Mexicana de Cultura A.C. for all their support. Specially we would like to mention Andrés Rodriguez, Juan Vera and David Jiménez for their wholehearted assistance.

References

1. Bikel, D.M., Miller, S., Schwartz, R., Weischedel, R.: Nymble: a high-performance learning name-finder. In: Proceedings of the Fifth Conference on Applied Natural Language Processing, pp. 194–201. Association for Computational Linguistics (1997)
2. BlueMessaging: About BlueMessaging. http://bluemessaging.com/about/
3. Borthwick, A.: A maximum entropy approach to named entity recognition. Ph.D. thesis, New York University (1999)
4. Carreras, X., Marquez, L., Padró, L.: Named entity extraction using adaboost. In: Proceedings of the 6th Conference on Natural Language Learning, vol. 20, pp. 1–4. Association for Computational Linguistics (2002)
5. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. J. Mach. Learn. Res. **12**, 2493–2537 (2011)
6. Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. Commun. ACM **51**(1), 107–113 (2008)
7. Figueira, A.P.F.: 5. the viterbi algorithm for HMMs - part i (2013). Available online at http://www.youtube.com/watch?v=sCO2riwPUTA
8. Forney Jr., G.D.: The viterbi algorithm: a personal history, April 2005. arXiv:cs/0504020, http://arxiv.org/abs/cs/0504020, arXiv: cs/0504020
9. Goddeau, D., Meng, H., Polifroni, J., Seneff, S., Busayapongchai, S.: A form-based dialogue manager for spoken language applications. In: Proceedings of the Fourth International Conference on Spoken Language ICSLP 1996, vol. 2, pp. 701–704. IEEE (1996)
10. Grishman, R., Sundheim, B.: Message understanding conference-6: a brief history. In: Proceedings of the 16th Conference on Computational Linguistics COLING 1996, vol. 1, pp. 466–471. Association for Computational Linguistics, Stroudsburg, PA, USA (1996)

11. Hurtado, L.F., Griol, D., Sanchis, E., Segarra, E.: A stochastic approach to dialog management. In: IEEE Workshop on Automatic Speech Recognition and Understanding 2005, pp. 226–231. IEEE (2005)
12. Jurafsky, D., Martin, J.H.: Speech and Language Processing. Pearson Education India, Noida (2000)
13. Karlin, S., Taylor, H.E.: A First Course in Stochastic Processes. Academic Press, New York (2012)
14. Kozareva, Z.: Bootstrapping named entity recognition with automatically generated gazetteer lists. In: Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop, pp. 15–21. Association for Computational Linguistics (2006)
15. Marcus, M.P., Marcinkiewicz, M.A., Santorini, B.: Building large annotated corpus english: the penn treebank. Comput. Linguist. **19**(2), 313–330 (1993)
16. Misu, T., Georgila, K., Leuski, A., Traum, D.: Reinforcement learning of question-answering dialogue policies for virtual museum guides. In: Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue, p. 8493. Association for Computational Linguistics (2012)
17. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Lingvist. Investig. **30**(1), 3–26 (2007)
18. Norris, J.R.: Markov Chains. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, New York (1999)
19. Capp, O., Moulines, E., Rydn, T.: Inference in Hidden Markov Models. Springer Series in Statistics. Springer, New York (2005)
20. Reese, S., Boleda, G., Cuadros, M., Padr, L., Rigau, G.: Wikicorpus: a word-sense disambiguated multilingual wikipedia corpus. In: Proceedings of 7th Language Resources and Evaluation Conference (LREC 2010). La Valleta, Malta, May 2010
21. Sang, T.K., Erik, F.: Memory-based named entity recognition. In: Proceedings of the 6th Conference on Natural Language Learning, vol. 20, pp. 1–4. Association for Computational Linguistics (2002)
22. Seneff, S.: Response planning and generation in the mercury flight reservation system. Comput. Speech Lang. **16**(3–4), 283–312 (2002)
23. Viterbi, A.: Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Trans. Inf. Theory **13**(2), 260–269 (1967)
24. Williams, J.D., Young, S.: Partially observable Markov decision processes for spoken dialog systems. Comput. Speech Lang. **21**(2), 393–422 (2007)
25. Zue, V., Seneff, S., Glass, J.R., Polifroni, J., Pao, C., Hazen, T.J., Hetherington, L.: Jupiter: a telephone-based conversational interface for weather information. IEEE Trans. Speech Audio Process. **8**(1), 85–96 (2000)