J Braz Comput Soc (2013) 19:135–146
DOI 10.1007/s13173-012-0095-1

ORIGINAL PAPER

Portuguese text generation using factored language models

Eder Miranda de Novais · Ivandré Paraboni

Received: 7 May 2012 / Accepted: 6 November 2012 / Published online: 24 November 2012

Abstract As in many other natural language processing (NLP) fields, the use of statistical methods is now part of mainstream natural language generation (NLG). In the development of systems of this kind, however, there is the issue of data sparseness, a problem that is particularly evident in the case of morphologically-rich languages such as Portuguese. This work presents a shallow surface realisation system that makes use of factored language models (FLMs) of Portuguese to overcome some of these difficulties. The system combines FLMs trained on a large corpus with a number of NLP resources that have been made publicly available by the Brazilian NLP research community in recent years, such as corpora, dictionaries, thesauri and others. Our FLM-based approach to surface realisation has been successfully applied to the generation of Brazilian newspapers headlines, and the results are shown to outperform a number of statistical and non-statistical baseline systems alike.

Keywords Natural language generation · Text generation · Surface realisation

1 Introduction

In natural language generation (NLG) systems, surface realisation is known as the task of mapping abstract sentence representations to a surface form, that is, a sequence of words, punctuation symbols etc., to be delivered to the document presentation system [1]. The input to the surface realisation module is a (mostly) language-independent representation of the meaning of the sentence, and the output is a word string in the target language.

In recent years, as in many other NLG applications, surface realisation systems have successfully relied upon statistical methods ranging from language modelling techniques [2,3] to full-blown probabilistic generation-space models [4] and grammar acquisition [5]. Our own work focuses on 2-stage, or generate-and-select NLG architectures as introduced in [2] and by others. Systems of this kind produce text from an abstract input representation by separating the generation space from decision-making, that is, by over-generating a large number of alternative surface realisations (often including non grammatical or ill-formed candidates) and subsequently selecting the most likely output string with the aid of a statistical language model.

Generate-and-select surface realisation shares many of the well-known advantages of statistical approaches to NLP, including lower development costs (e.g., by not requiring corpus annotation etc.) and language independency. However, it also shares one of its main weaknesses, namely, the need for large amounts of training data to compensate for data sparseness.

In statistical language modelling, data sparseness is particularly acute in the case of morphologically rich languages. For instance, a Portuguese language model will typically require much larger amounts of training data if compared to, e.g., an English language model, in order to achieve comparable results [6]. Moreover, English NLG systems may also benefit from state-of-art resources such as the 1-trillion words Web 1T 5-gram corpus,[1] whereas the largest publicly

FAPESP grant 2009/08499-9.

E. M. de Novais · I. Paraboni (✉)
School of Arts, Sciences and Humanities,
University of São Paulo (EACH/USP), Av. Arlindo Bettio,
1000 São Paulo, Brazil
e-mail: ivandre@usp.br

E. M. de Novais
e-mail: eder.novais@usp.br

[1] http://www.ldc.upenn.edu/Catalog/.

available corpus of Brazilian Portuguese at the time of this writing was the 32-million words *NILC* corpus [7].

As a means to overcome some of these difficulties, in this work we address the use of factored language models (FLMs) [8] in the development of a shallow surface realisation system for the Brazilian Portuguese language. In doing so, we would like to show that FLMs may outperform standard n-gram models in a traditional generate-and-select NLG architecture, and may represent a viable solution to the problem of data sparseness in morphologically rich language generation.

Besides the inherently greater expressive power of FLMs over standard n-gram counts as discussed later, our models are trained on a considerably larger (142-million words) corpus of Brazilian Portuguese texts, and are combined with a number of nonstatistical NLP resources developed by the Brazilian research community in recent years, such as dictionaries [9] and thesauri [10], making a surface realisation system that represents, to our knowledge, the first of its kind. Our FLM-based approach to surface realisation is applied to the generation of newspapers headlines, and the results are shown to outperform statistical and nonstatistical baseline systems alike.

The rest of this paper is organised as follows. Section 2 briefly reviews related work in the field, and Sect. 3 describes the main FLM-based approach that is the focus of the paper. Evaluation work is presented in Sect. 4, and its results are discussed in Sect. 5. Section 6 presents final remarks.

2 Related work

Surface realisation can be viewed as the task of mapping abstract sentence representations to word strings in a target language. What counts as an input representation may, however, vary widely across systems, and the lack of a standard on how a surface realisation system should be defined—or more specifically, on what makes its input specification—has long been the subject of debate in the NLG field [1]. The work in [2], for instance, considers its own sentence representation in the form of labelled directed graphs. The work in [11], by contrast, takes as an input logical forms representing the meaning of the sentence. More importantly, different systems may consider more shallow (i.e., closer to the surface form) or deeper input representations, which implies different functionalities.

As a first initiative towards standardisation in the field, the first *Surface Realisation Shared Task* [12] intended to evaluate English surface realisation systems based on the same input data (abstracted from the *WSJ* corpus). The shared task was accomplished by five participants, being three statistical and two symbolic systems. Results discussed in [12] show that the statistical approaches were overall more successful. Although the goal of the shared task was to enable a direct, clear comparison among the participant systems, a number of input-specification issues remain to be solved. For a discussion on these difficulties and future improvements, see [13].

Out of the five participants in the shared task, three systems deserve special mention.[2] The work in [16] is a statistical surface realisation engine that makes use of dependency-based n-gram models (i.e., as opposed to standard n-gram counts) to exploit structural information and linguistic features to constrain the generation space. The work in [17] is also a statistical generator, acquiring all relevant information for the surface realisation task from corpora. This information comprises a localised tree model, a morphological dictionary and a trigram language model. The only nonstatistical shallow generator in the competition was the work in [18], which makes use of unification-based grammars. It should be pointed out however that all these systems produce English text from data derived from the *WSJ* treebank, and, therefore, a direct comparison to our Portuguese text generator is not possible.

One useful way of distinguishing different approaches to surface realisation is proposed in [19], in which the task is viewed as divided into two rather independent components: a tactical generator in charge of defining the mappings from semantic inputs to morphosyntactic structures, and an operational generator that, once all tactic decisions have been made, performs the appropriate morphological tasks and sentence linearisation as a string in the target language [19]. Tactical generation tends to be more domain- or application-dependent, whereas operational generation tends to be more language-dependent. However, the distinction is not always crisp, and many systems still perform both tasks [19].

In [19], operational or shallow realisation systems are referred to as *surface realisation Engines*, and a system called *SimpleNLG* is introduced. The system is presented as a Java library for creating and manipulating English sentences. Generating text with *SimpleNLG* consists of invoking a series of methods to specify every sentence constituent, how they should be inflected, combined etc.

Similarly to [19], our present system is an instance of a surface realisation engine. However, our system is distinct from *SimpleNLG* in a number of ways. First, we consider an explicit input specification adapted from [12], from which a sentence is produced by making a single call to the language generator. In *SimpleNLG*, by contrast, a sentence is produced incrementally by making a series of calls to the system. Second, *SimpleNLG* is a rule-based surface realisation engine, whereas our present work combines rule-based and statistical methods. Third, *SimpleNLG* generates English text, whereas

[2] The other two participants, the systems described in [14] (using a graph-based parsing algorithm) and [15] (a grammar-based chart realiser), focused on generation from deep input representation.

we focus on Portuguese (and to this end both systems encode language-dependent rules).

Regarding the statistical approach under consideration, our present work follows the generate-and-select NLG architectures introduced in [2,20] and others. Systems of this kind generate surface strings from some abstract input representation in two stages: first, a large number of possible candidate output strings are generated from the given input, and then the most likely output string is selected with the aid of a statistical language model. However, existing work on 2-stage surface realisation has been traditionally based on n-gram models only, whereas the present work makes use of FLMs [8] instead. To our knowledge, our system is the only FLM-based surface realisation engine to date, and the only of its kind that has been designed for the Portuguese language.

Our current system is the final product of a series of experiments on surface realisation for Brazilian Portuguese. The work in [21] described a number of experiments on n-gram filters applied to individual surface realisation tasks, namely, lexical choice, ordering of noun modifiers and verb-complement agreement. These experiments were performed over purpose-built sentences and their focus was on the identification of critical surface realisation tasks from the perspective of a 2-stage generation approach based on n-gram models.

The n-gram approach in [21] was shown to perform poorly in three tasks: verb phrase lexicalisation, the ordering of noun modifiers and verb-complement agreement (mainly in passive voice). These issues were revisited in [6] with the introduction of FLMs applied to text generation in Brazilian Portuguese. The work in [6] was the first of this kind to propose FLMs that outperformed standard n-gram models for Portuguese text generation. However, the work in [6] was still limited in the sense that it was not applied to the generation of whole sentences taken from a corpus of actual language use, and by using large, computationally expensive language models that took gender and number factors into account, and which we have presently improved on.[3]

Finally, the work in [22] is the closest to the present discussion, introducing a preliminary—but still purely statistical—version of our system and the training and test data sets that we have presently reused. However, the system presented in [22] was still lacking many of the current functionalities, including the rule-based methods to limit over-generation and the use of a thesaurus to exploit word synonymy. The current system, by contrast, outperforms its early version in [22], but it is of course more language-dependent.

[3] As discussed in Sect. 3.4, given that our present training corpus is several times larger than the one used in [6] a direct comparison between the computational efficiency of these approaches is not possible.

3 Current work

3.1 Overview

The present work concerns the development and evaluation of a shallow surface realisation system for the Brazilian Portuguese language based on FLMs. Systems of this kind are mostly applicable to text-to-text generation tasks such as text summarisation [23], simplification [24] and others, but may also be embedded in a deeper generation framework [1]. Figure 1 shows the system architecture, in which grey boxes represent its three main modules: symbolic pre-processing, symbolic over-generation and statistical candidate selection. White boxes represent external knowledge sources as discussed below.

The system takes as an input an under-specified abstract sentence representation (cf. Sect. 3.2). In order to establish agreement between sentence constituents, missing input values are adjusted and/or complemented with the aid of a lexical database described in [9]. The result is a fully specified sentence representation taken as the input to the next module, which over-generates alternative realisations (cf. Sect. 3.3) based on linearisation constraints (also obtained from lexical information) and, optionally, synonymy information taken from a thesaurus of Brazilian Portuguese [10]. Finally, the set of possible candidates is submitted to a language model filter and the most likely output sentence is selected. As discussed in Sect. 3.4, the main difference between this and a more traditional generate-and-select architecture (e.g., [2]) is the use of FLMs in the selection task, and not standard n-gram counts. Details of each of these steps are discussed in the following sections.

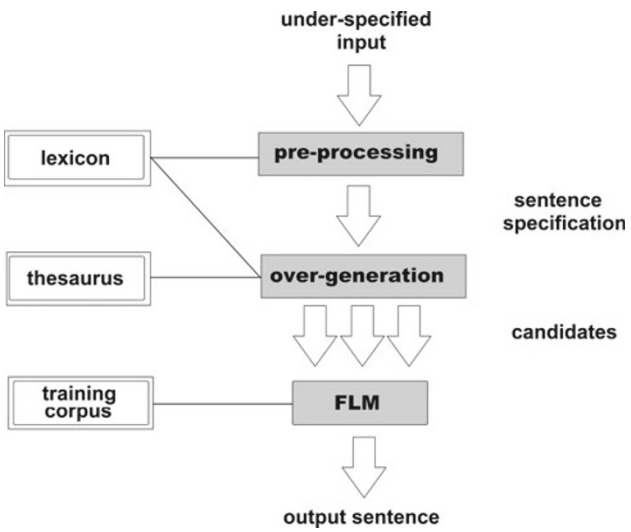

Fig. 1 System architecture

3.2 Input processing

The input to the system is a shallow, possibly under-specified sentence representation similar to the specification considered in the *Surface Realisation Shared Task* [12]. As a case study, however, the input specification is presently more limited in the sense that we will only consider sentences in active voice according to a standard < NP-VP-NP > template order.[4]

The input is represented as a sentence tree in which all content words are assumed to have been previously selected following some lexical choice policy (e.g., [25,26]). Thus, the task to be accomplished by the system mainly consists of providing missing input information, establishing constituents agreement and word order in the target language (in our case, Brazilian Portuguese).

The following is a simplified example of input specification for a target sentence *'A cantora americana Christina Aguillera não terá um programa de rádio semanal'* or 'The American singer Christina Aguillera will not have a weekly radio program'. The example is rendered in Prolog-like clauses generated automatically from a tagged corpus using a simple conversion utility.

sentence(u_1, [agent(s_1), action(v_1), patient(p_1), punctuation['.']]).
concept(s_1, obj, [s, ['americano'], head('cantor'), ['christina=aguilera'], def, f]).
concept(p_1, obj, [m, s, rel(['de', 'rádio']), semanal, head('programa'), indef]).
concept(v_1, act, [ind, fut, head(['ter']), vfin, ['não']]).

Example 1 Input specification for sentence u_1

The top-level sentence representation is the *sentence* clause, in this case representing a fixed order *Agent-Action-Patient-Punctuation* sentence template. *Agent* and *Patient* terms are defined as object concepts (s_1 and p_1) to be realised as noun phrases (NPs). *Action* terms are defined as act(ion) concepts (v_1) to be realised as verb phrases (VPs). Both *Patient* and *Punctuation* terms are optional.

Concepts to be realised as noun phrases (either filling in *Agent* or *Patient* positions in the sentence template) are unordered lists of nouns, proper names, adjectives and prepositional phrases (PPs). The PP surface forms may be computed recursively during generation but, for reasons of computational efficiency, these constituents are presently assumed to be computed in advance. In other words, they are assumed to appear already in their final surface form (regarding both inflection and order) in the input specification. Although this may in principle be seen as a limitation of our approach, in practice nearly all (over 99 %) noun phrases in our test corpus (newspapers headlines, cf. Sect. 4) have at most two PP modifiers each, making their computation rather trivial. Concepts to be realised as VPs (filling in an *Action* position) are sets of verbs with accompanying prepositions and adverbial modifiers.

Each NP or VP term has a head constituent, to which all other elements are subordinated. The head constituent is defined solely for the purpose of providing information for other constituents when necessary, as discussed later, but it does not follow that the head constituent needs to be defined according to a strict linguistic theory. Typical head constituents for NPs are nouns and proper names. For VPs, the head constituent is usually the main verb of the sentence.

Given an input specification as above, the system starts by computing any missing information deemed necessary for the agreement and linearisation tasks from left to right. Missing values are provided by examining *Agent*, *Action* and *Patient* terms individually, in that order. We assume that the information provided for the *Agent* term drives the generation process and, if necessary, will overlap any conflicting information provided by the other terms. For instance, if the underlying application provides conflicting instruction to produce a sentence whose *Agent* term should be singular but whose *Action* term should be plural, *Action* will be adjusted to singular, and not the other way round.

Starting from the *Agent* term, the system will use the gender and number provided or, if necessary, obtain these values from any nonambiguous constituent within the *Agent* term. A constituent is considered free from gender/number ambiguity when it occurs in a single gender/number form in the lexical database implemented in [9]. Within a given term *t*, finding a nonambiguous constituent for which there is only one possible gender *g* and number *n* value implies that all constituents within *t* must have the same gender *g* and number *n* values. In other words, nonambiguous gender/number values, if available, will be taken to be the obligatory gender/number values for all constituents within the same term.

For instance, concept s_1 in the previous example includes number information ('s' for 'single'), but no gender. In order to determine the gender of the NP, the system will make use of lexical information to examine proper names (which are less likely to have more than one gender/number form), followed by nouns and adjectives, if necessary. If the proper name 'Christina Aguillera' is represented in the dictionary only in female form, this will determine the gender of all constituents of the *Agent* term. If no constituent is represented in a single form, default values provided by the underlying application are attempted. Finally, if no default value is applicable (e.g., if some constituents can only be realised in singular form,

[4] On the other hand, the more complex sentence structures considered in [12] require a richer input representation. This includes, for instance, syntactic edges and other features that are not presently required.

r1 Punctuation marks are always placed at the end of the sentence.

r2 The sentence follows the order *Agent*, *Action*, *Patient* of terms.

r3 Determiners are placed always at the beginning of an NP.

r4 Compound proper names are taken to be in the correct word order.

r5 PP modifiers are taken to be in the correct word order.

r6 NP determiners cannot be followed by PP attachments.

Fig. 2 Linearisation rules

and others in plural) the input is considered inconsistent and no text is generated.[5]

Once gender and number values of the *Agent* term have been computed, these are enforced for the *Action* term as well and, in case of copula verb usage (e.g., 'She is pretty') also extended to the *Patient* term. This is necessary because the Portuguese language (unlike, e.g., English) requires subject-complement agreement in these cases, as in '*Ela é bonita*', in which the subject ('*Ela*', or 'she') agrees in gender and number with the complement ('*bonita*', or 'pretty').

After establishing complete agreement within and between terms, additional features (verb tense, mode, etc.) are determined using the same principles of inheritance and default values provided by the underlying application. The next step consists of a standard over-generation approach [2,28,29] in which all potentially valid permutations of the term set are produced.

3.3 Over-generation

Given a complete input sentence specification (i.e., with all relevant gender and number values in agreement, etc.) over-generation consists of producing a set of possible linear orderings of the sentence constituents. In order to limit the number of candidates under consideration, however, we define valid permutations according to a number of simple linearisation rules. These rules are intended both to reduce the computational costs of the selection task (cf. next section) and also to increase the overall probability of finding the most adequate output string, but are by no means to be seen as a substitute for more structured solutions such as grammar acquisition (e.g., [5]). These linearisation rules are summarised in Fig. 2.

Except for the punctuation and sentence template rules (*r1* and *r2* above), all linearisation rules are applicable to NPs. This choice is due to the fact that, in our system, the sentence under generation follows a previously chosen template format, and that VPs (at least as seen in our test data on newspapers headlines, described in Sect. 4) tend to have a small number of constituents (usually one or two verbs, plus one occasional adverbial modifier) and are generally well covered by the language model filters. It was not necessary, for instance, to define a rule to prohibit VP alternatives containing a misplaced negative adverb as in *'terá não'* ('will not have'), which are considered in 3 % of our test sentences. Ungrammatical constructions of this kind are correctly filtered out by the language models. In other words, as observed in [6], it is the NP over-generation task that presents most opportunities for improvement.[6]

An example of application of NP linearisation rules works as follows. Rule *r3* guarantees that definite and indefinite determiners are always placed at the beginning of an NP, which rules out illegal candidates such as *'cantora a'* (or '*singer the'). Rule *r4* prohibits the permutation of proper names as in '*aguilera christina', which are assumed to be always presented in the correct linear order. As argued in [12], computing the word order of proper names is not an actual surface realisation task. Similarly, rule *r5* determines that PP modifiers as in *'de rádio'* are to be taken as a single term, that is, they are not subject to permutation as in *'rádio de'*.

Finally, sentences in our test data often combine several PP or adjectival modifiers into a single NP, as in *'o valor integral do aumento salarial'* ('the pay rise full amount'), leading to a potentially large number of NP permutations. Some of these alternatives, however, are clearly ungrammatical, as in *'*do aumento salarial o valor integral'*, or, '*of the pay rise the full amount'. For that reason, over-generation is limited by rule *r6*, which discards all NP permutations (in both subject and object position) whose determiner is immediately followed by a preposition (and which are, at least in Portuguese, unacceptable).

The benefits of these simple NP linearisation rules to both computational efficiency and output accuracy are self-evident. On the other hand, adding handcrafted rules to an otherwise purely statistical approach may always incur a well-known cost, namely, a loss in language-independency. As we will argue in Sect. 5, however, we believe that at least in the present case the benefits of linearisation rules may

[5] The present use of default values provided by the application is not to be confused with the inference task in inheritance-based grammars (e.g., [27]).

[6] Alternative approaches to the task of ordering noun modifiers are discussed in [30,31].

possibly outweigh any loss of this kind. In our test data, for instance, the use of one single rule (*r6*, prohibiting NP determiners followed by prepositions) has decreased the candidate set produced during over-generation from 37,462 sentences to 19,869 sentences, that is, a 47 % reduction. Although this is much smaller than the 87 % reduction estimated in [22], in which a more liberal over-generation strategy was considered, linearisation rules are still likely to improve results, particularly when combined with more expressive FLMs (see Sect. 3.4).

All constituents in all orderings are replaced by their inflected surface forms with the aid of the lexicon in [9], producing a list of candidate output strings for the given input. For instance, from the *Action* constituent set in the previous example we may obtain two candidate word strings: *'terá não'* and *'não terá'* ('will not have'). Similarly, we would obtain six output candidates from the *Agent* constituent set, and four candidates from *Patient*,[7] making (6 * 2 * 4 =) 48 candidate sentences in total. Some of these alternatives are illustrated below, among which sentence 48 happens to be the expected output.

1 *A cantora americana Christina Aguilera terá não um semanal programa de rádio.*
2 *A americana cantora Christina Aguilera terá não um semanal programa de rádio.*
3 *A Christina Aguilera cantora americana terá não um semanal programa de rádio.*
...
48 *A cantora americana Christina Aguilera não terá um programa semanal de rádio.*

Example 2 Candidate set for the input in Example 1

Optionally, a candidate set as above may be further expanded by considering synonymy as well. In our work this is implemented with the aid of the Brazilian Portuguese thesaurus presented in [10]. Additional alternatives are generated by replacing head constituents and their modifiers by synonymous words, and then following the same over-generation rules described above. However, the use of synonyms is offered simply as an additional feature of our system, and it is left entirely to the underlying application to decide whether to use it or not. As suggested in [25,26] and also by our preliminary experiments in [21], using synonyms in a principled way remains an open research question, and in order to prevent lexicalisation issues from obscuring the comparison between language models, our evaluation work described in Sect. 4 will not cover lexicalisation.

3.4 Language modelling

In order to single out the desired output sentence from an over-generated candidate set, 2-stage generation systems [2,3,20,28,29] make use of statistical language models to compute the most likely output sentence. Language models of this kind are usually based on n-gram-counts, which are known to produce satisfactory results at least for less-inflected languages such as English. However, in the case of our target language—Brazilian Portuguese—we have to deal not only with much greater lexical variation (making data sparse) but also with the lack of sizeable training data. In what follows we attempt to minimise both difficulties by (a) making use of more expressive language models and (b) by making use of a larger training corpus.

From the language modelling perspective (a), we address the issue of data sparseness by replacing standard n-gram models for Factored Language Models (or FLMs, cf. [8]). FLMs generalise the n-gram approach by incorporating information from various sources (and not only word counts). FLMs are widely used by the speech research community but, to our knowledge, their benefits have been seldom applied to text generation.[8]

In the FLM approach, each word w_t is represented as a bundle of k parallel factors $\{f_t^1, f_t^2, \ldots f_t^k\}$ that may convey any word-related information such as classes, roots, semantic features, etc. Thus, when no exact n-gram match is found in the training data (as it is often the case in morphologically-rich languages), an FLM will allow us to circumnavigate the issue of data sparseness by taking into account such alternative knowledge sources (and not simply word counts).

The design of an FLM requires the definition of a set of individual factors and a strategy to produce an optimal statistical model over them. When there is insufficient data to fully estimate a higher-order condition, a standard n-gram model would simply backoff from a model of order n to an order n-1. FLMs, by contrast, may simultaneously drop one or more variables associated to each factor, and take different factors into account at the lower level.

For instance, when a particular trigram does not occur (a certain number of times) in the data, the model may consider probabilities given by the lemma of the previous word L_{t-1} and the part-of-speech of the word before that P_{t-2}, and that these probabilities should be combined in a particular way to estimate the overall probability of the current word W_t. Thus, even the simplest FLMs allow a large number of possible backoff paths, making directed *backoff graphs* as discussed in [8]. As we shall see, this may have significant advantages over simply backing-off down to the bigram or unigram level.

We consider bigram and trigram models taking into account either word and lemma factors (*WL* models) or word, lemma and part-of-speech factors (*WLP* models) only. We will call these our *2WL*, *3WL*, *2WLP* and *3WLP* models, respectively. These models represent the best tradeoff between computational efficiency and output accuracy that

[7] Recall that alternatives beginning with a PP as in *'de rádio'* are disregarded.

[8] An exception is the work in [32] regarding the acquisition of NLG grammars aided by FLMs.

we could obtain over a large number of experiments in text generation. Some of the alternatives to these models have been described in previous work (see Sect. 2) and include both models of higher order and those using additional factors. For instance, the experiments in text generation described in [6] made use of FLMs that considered gender and number factors as well. However, the impact of those factors on output text quality has been found to be small if compared to the increase in computational costs, and for that reason gender and number have been presently disregarded.

The proposed models are summarised in Fig. 3. Unless specified otherwise, all models were built using *SRILM* [33] and using the default tool parameters and a back-off strategy as follows. The first model to be attempted is always the complete model (i.e., taking all available factors into account). If necessary, the model will back-off to lower levels by dropping one node at a time, starting from the most distant parent node, and discarding the factors *W*, *L* and *P* in this fixed order, that is, from more to less informative factors as suggested in [8].[9]

We assume a minimum count (the *gtmin* parameter in *SRILM*) of 1 to establish a match,[10] and we compute model probabilities by interpolation with the lower levels. For details on the mathematical background and design of FLMs and parallel back-off, see [8].

Let us consider the above *2WLP* model as an example. The first backoff-graph node in *2WLP* corresponds to the complete model that takes all available factors into account, that is, the probability of a word W_t is given by the combined probabilities of the previous word (W_{t-1}), lemma (L_{t-1}) and part-of-speech (P_{t-1}) factors. If no instantiation of these three values exists (i.e., if these three particular *W*,*L* and *P* values do not co-occur in the data), the model will backoff by dropping the *W* factor, that is, by attempting to estimate probabilities based on the previous *L* and *P* factors alone. If that fails once again, the model will still attempt to estimate the word probability based on the previous part-of-speech factor only, the underlying assumption being that P_{t-1} may still provide some (even if rather weak) hint at the probability of the current word when no other source of information is available. Thus, if a bigram as, e.g., 'did fail' does not occur in the data, we may still estimate some probability by considering the more general form 'do fail' or even '<verb> fail' if necessary.

We leave to Sect. 4 to discuss to which extent the use of FLMs improve results in the text generation task, but first we shall briefly consider the corpus perspective (b). Since the

$2WL:$

$$p(W_t|W_{t-1}, L_{t-1}).$$
$$p(W_t|L_{t-1}).$$

$3WL:$

$$p(W_t|W_{t-1}, L_{t-1}, W_{t-2}, L_{t-2}).$$
$$p(W_t|W_{t-1}, L_{t-1}, L_{t-2}).$$
$$p(W_t|W_{t-1}, L_{t-1}).$$
$$p(W_t|L_{t-1}).$$

$2WLP:$

$$p(W_t|W_{t-1}, L_{t-1}, P_{t-1}).$$
$$p(W_t|L_{t-1}, P_{t-1}).$$
$$p(W_t|P_{t-1}).$$

$3WLP:$

$$p(W_t|W_{t-1}, L_{t-1}, P_{t-1}, W_{t-2}, L_{t-2}, P_{t-2}).$$
$$p(W_t|W_{t-1}, L_{t-1}, P_{t-1}, L_{t-2}, P_{t-2}).$$
$$p(W_t|W_{t-1}, L_{t-1}, P_{t-1}, P_{t-2}).$$
$$p(W_t|W_{t-1}, L_{t-1}, P_{t-1}).$$
$$p(W_t|L_{t-1}, P_{t-1}).$$
$$p(W_t|P_{t-1}).$$

Fig. 3 Language models under consideration

training data used in previous work (e.g., [6]) was deemed insufficient even in the case of standard bigram models, we now intend to use more data to take full advantage of the FLM approach. For that reason, the present language models were built from a 142-million words corpus of Brazilian Portuguese described in [34], which included both the original *NILC* corpus [7] and an additional collection of full articles from the on-line 2006–2011 editions of the *Folha de São Paulo* newspaper[11] and the *Veja* magazine.[12]

The training corpus was POS-tagged using the Portuguese tagger available from the *LACIO-WEB* website project[13] and *MXPOST*.[14] Additionally, gender and number tags (required for the pre-processing agreement task, but not used in the actual language models) were taken from the Brazilian Portuguese lexical database described in [9]. Tagging errors identified during the preparation of the test data (particularly in the case of proper names, cf. next section) were also corrected in the training data to ensure consistency, but the rest of the training corpus was left otherwise unaltered, assuming that the statistical approach should accommodate noisy data.

[9] Less conventional back-off strategies were also attempted in a pilot experiment, but results turned out to be below those presently reported (cf. Sect. 4).

[10] Pilot tests with higher threshold values in [21] showed lower accuracy rates in related tasks.

[11] http://www.folha.com.br.

[12] http://www.veja.com.br.

[13] http://nilc.icmc.sc.usp.br/nilc/tools/nilctaggers.html.

[14] http://www.inf.ed.ac.uk/resources/nlp/local_doc/MXPOST.html.

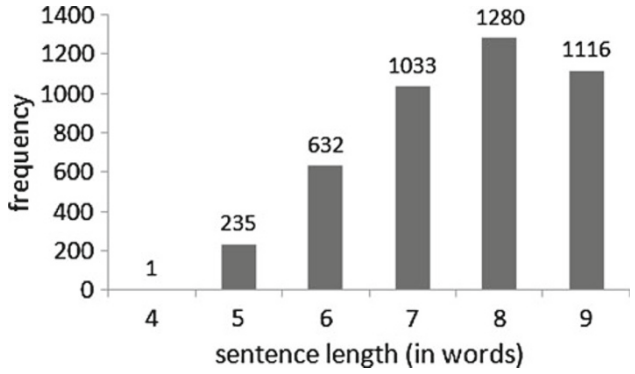

Fig. 4 Sentence length distribution in the corpus test

4 Evaluation

4.1 Test data

We consider a test corpus comprising a separate collection of 4,297 randomly selected online *Folha de São Paulo* newspaper headlines from the year 2009, of up to 9 words in length each (7.6 words on average). The actual distribution is illustrated in Fig. 4.

All sentences in the test data set are in the form *Agent-Action-Patient*. The *Patient* term was not compulsory, but it turned out to occur in all but 18 sentences. In comparison with the training corpus described in the previous section (containing full articles), test sentences are often simpler, and one may in fact ask why we used a training corpus more powerful than necessary for the task. Our reasons are twofold: first, collecting a sufficiently large number of individual headlines for training purposes may be impractical; second, as future work we intend to reuse the same models in the generation of newspapers sentences in general, and not only headlines.

Our *Input* data set was built as follows. First, the corpus was tagged with POS, gender and number information and manually verified for correctness. This was particularly necessary in the case of proper names, which occur in large numbers in the newspapers domain, and were often tagged as nouns, etc. In addition to that, some instances of verbs in the participle tense were incorrectly tagged as adjectives, and in a few cases gender/number information taken from the dictionary had to be corrected as well. This revision was, however, informal in the sense that we did not seek to produce a 100 % accurate corpus (for instance, we did not seek to achieve agreement between judges) but simply to identify common errors that could affect our evaluation work.

All tagging errors identified in the test corpus were also corrected in the training corpus (described in the previous section) by using a purpose-built tool. Briefly, the tool takes as an input the list of corrections previously made, and then performs a simple search-and-replace operation on the training corpus, rewriting any text segment containing the identified errors provided that there was no risk of ambiguity. For instance, we did not perform any conversion from noun 'machado' ('axe') to proper name as this could result in a tagging error, even though 'Machado' is a common Portuguese name. Only after these adjustments were completed, the language models described in the previous section were generated.

Each test sentence was converted into an abstract sentence representation by performing two tasks: (1) every content word was replaced by its lemma and (2) the constituent ordering of NP and VP modifiers was randomised. Similar techniques for abstracting NLG input data from the expected output strings were applied, for instance, in the data preparation for the *First Surface Realisation Challenge Task* [12]. The resulting abstract sentence representations are similar to the previous Example 1 in Sect. 3.2.

The actual word strings in the corpus are taken to be our *Reference* set, against which we intend to compare (therefore intrinsically) the output produced by 10 alternative generation strategies: the four FLMs (*2WL*, *3WL*, *2WLP* and *3WLP*) discussed in the previous section, and six baseline systems as follows.

4.2 Procedure

For the purpose of evaluation, one may consider at least three lines of investigation: (1) how the statistical module performs with and without linearisation rules; (2) how FLMs compare to word n-grams; and (3) how the system as a whole compares to others.

The role of linearisation rules has been addressed to some extent in our previous work [6] in the context of NP surface realisation,[15] in which the use of FLMs alone was found to be insufficient for the task. Using 2WL and 3WL models, the work in [6] showed that candidate selection from an unconstrained set of possible realisations obtained accuracy rates of 0.70 and 0.60, respectively. On the other hand, by applying basic rules to reduce the candidate set, the same models obtained accuracy rates of 0.85 and 0.80. These results first suggested to us that the FLM approach could be improved on by using this form of pre-selection, and for that reason we presently do not seek to answer (1) directly, that is, we will simply take the benefits of linearisation rules for granted.[16]

[15] Recall also that NP linearisation is the most challenging subtask for our system.

[16] Note also that reducing the candidate set cannot lead to *lower* accuracy.

Question (2)—representing the main claim that we intend to verify—is sufficiently straightforward: keeping other variables unaltered (namely, using always the same linearisation rules), we will compare different versions of our system using various language models (FLM and word n-gram alike). To this end, we will consider three baseline systems that use the same over-generation module but, instead of FLM filters, use ordinary 2-, 3- and 4-gram models called *2W*, *3W* e *4W*. In other words, keeping all other system features unchanged except for the actual language model, we would like to show that the FLM approach outperforms these word n-gram models, as suggested by some of our previous experiments in [22].

It is question (3) that poses the most significant challenge to our evaluation work (and indeed to the evaluation of any language-dependent task). As discussed in Sect. 2, differences in target language and input specification make a direct comparison to, e.g., the participants in the *Surface Realisation Shared Task* [12] or to *SimpleNLG* [19] inappropriate. Thus, in what follows we will consider a number of simple, nonstatistical baseline systems, and also a rule-based realisation engine developed as an independent project [34].

Two of our nonstatistical baseline systems are trivial: a *Random* strategy that selects a random permutation from the set of possible candidate sentences, and a *Left-to-Right* template-based approach that simply fills in the sentence template with matching constituents in the same order as they occur in the randomised input specification.

As a third and more robust alternative, we will also consider a *Rule-based* strategy described in [34], which makes use of Portuguese grammar rules to enforce constituents agreement and sentence linearisation. This system represents a complete realisation engine and, for the purpose of the present evaluation task, it may be considered a reasonable approximation to *SimpleNLG* [19] in the sense that both systems are capable of generating common sentences in their target languages (Portuguese and English, respectively) in similar step-by-step fashion.[17] More importantly, our Rule-based system takes as an input the same representation required by our current approach, and generates text in the same target language (i.e., Portuguese), which allows for a direct, meaningful comparison between statistical and nonstatistical approaches.

Each of the 10 systems took as an input the same *Input* data set and produced a corresponding *System* set of 4,297 output strings. The main purpose of the evaluation was to measure the closeness between each *System* and *Reference* sentence pairs, and for reasons discussed in Sect. 3.3 none of the systems was set to consider synonymous words during over-generation.

The actual comparison between *System* and *Reference* sentence pairs was carried out by using four standard string metrics (which are expected to correlate): Accuracy or string match (exact word string match equals 1, or 0 otherwise); Edit-distance (i.e., the number of insert, delete and replace operations required to make both strings identical); and the Machine Translation (MT) metrics BLEU [35] and NIST [36],[18] both of which measure n-gram overlap between *Reference* and *System* (BLEU scores range from 0 to 1, and NIST scores have no upper limit[19]).

A word of caution regarding the interpretation of Accuracy scores: none of the systems under evaluation is currently able to make an informed distinction between multiple equally acceptable alternatives, as in *'programa de rádio semanal'* vs. *'programa semanal de rádio'*, both of which translate to 'weekly radio program'. In these situations, Accuracy will heavily penalise all systems that make the 'wrong' choice, that is, those which do not select the exact word string that happens to occur in the *Reference* set. Edit-distance, BLEU and NIST scores, by contrast, represent more fine grained metrics of closeness between *System* and *Reference* sentences, and are therefore to be preferred. This limitation notwithstanding, we will follow common practice in the field (e.g., [37]) and provide Accuracy results for illustration purposes.

4.3 Results

Table 1 summarises our results. Closer proximity to the *Reference* sentence is indicated by *higher* Accuracy, NIST and BLEU values, but *lower* Edit-distance.

Given that BLEU and NIST evaluate each system output set as a whole (and not individual sentence pairs), and given that Accuracy is simply a binary condition representing whether each string pair is identical, we performed one-way ANOVA for independent samples over Edit-distance values, which were shown to have normal distribution and homogeneity of variance. This was followed by the Tukey HSD test ($\alpha = 0.05$).

Results showed significant differences between the systems ($F(9, 43) = 400.38$, MSE $= 51.74$, $p < 0.001$). The identified homogeneous subsets are shown in Table 2.

[17] This is not to say, however, that these systems are functionally equivalent, since differences in target language do not allow a direct comparison between the two.

[18] Despite being originally proposed in the MT field, BLEU and NIST have been widely applied to the evaluation of a number of NLG surface realisation tasks as well [12,37,38].

[19] Unlike BLEU, NIST tends to favour less frequent and possibly more informative n-grams. cf. [36].

Table 1 Results

System	Accuracy	Edit-dist.	NIST	BLEU
Random	0.48	8.77	11.15	0.46
Left-right	0.62	5.52	14.58	0.81
Rule-based	0.69	4.55	14.97	0.85
2W	0.74	4.07	14.84	0.88
3W	0.75	3.84	14.83	0.88
4W	0.75	3.84	14.85	0.88
2WL	0.87	2.10	15.02	0.94
3WL	0.88	1.95	15.05	0.95
2WLP	0.89	1.75	15.04	0.94
3WLP	0.90	1.65	15.07	0.95

Table 2 Homogeneous subsets for Edit Distance values

Strategy	Avg. edit-dist.	Subsets				
3WLP	1.65	A				
2WLP	1.75	A				
3WL	1.95	A				
2WL	2.10	A				
4W	3.84		B			
3W	3.84		B			
2W	4.07		B			
Rule-based	4.55			C		
Left-right	5.52				D	
Random	8.77					E

Systems which do not share a letter differ significantly at $\alpha = 0.05$

5 Discussion

The results of the evaluation work show that the systems making use of FLMs outperform all standard n-grams models (*2W*, *3W* e *4W*) and nonstatistical (Random, Left-right and Rule-based) baseline systems alike. According to Table 2 there was no statistical difference within the FLM or within the n-gram groups. In the case of FLMs, this in principle favours the simplest, most computationally efficient *2WL* model, and in the case of n-grams favours the *2W* model. However, this is not to say that trigram models are not useful for the present task, or that those models still suffer from data sparseness. The high homogeneity in the results may be simply an effect of the kind of test data used in the evaluation: bigram models seem sufficient for the generation of short newspapers headlines, but for longer sentences a higher-order model may be called for. Had we considered longer sentences, it might have been possible to observe greater differences within the FLM and n-gram groups, as the progression of Edit-distance, BLEU and NIST scores in previous Table 1 seems to suggest.

The present results for the FLM approach are also superior to all of our own previous work. For instance, in a previous, purely statistical version of our system [22] with no associated rules to limit the number of output candidates, Edit-distance scores ranged from 2.69 for a *2WL* model to a maximum 1.84 for a *3WLP* model using the current test data. In the present case, by contrast, our best performing model achieved 1.65 edit-distance. As expected, the inclusion of the linearisation rules described in Sect. 3.3 does improve results, arguably at the cost of a certain loss in language-independency.

All systems that consider some kind of statistical model (FLMs and n-grams alike) outperform our admittedly simple nonstatistical baseline systems. The combination of rules and language models, even in the simplest *2W* approach, outperforms the Rule-based system as well. This result had not been observed when considering the purely statistical approach in [34], in which Rule-based surface realisation still outperformed standard n-gram models.

6 Final remarks

This paper presented a novel application of factored language models (FLMs) in the field of text generation, in which FLMs are intended to overcome data sparseness, an issue that is particularly prevalent in morphologically-rich language processing. We described a 2-stage shallow surface realisation system for Brazilian Portuguese generation that takes as an input an abstract sentence representation and over-generates multiple candidate sentences with the aid of a small set of language-dependent linearisation rules. Candidate selection is performed in a language-independent fashion by using a statistical language model trained on a large corpus of Brazilian newspapers articles, after which the language model filters out unsuitable alternatives and selects the most likely candidate as the output sentence.

The system was applied to the generation of newspapers headlines, in which sentences extracted from online articles were regenerated by a number of variations to our basic approach, and also by several statistical and nonstatistical baseline systems. Results show that using FLMs for Portuguese text generation represents not only a novel application of these models, but also suggest that the FLM-based approach may be indeed superior to the use of word n-gram models for this task.

On the other hand, although not presently discussed, the kinds of FLM considered in this paper are significantly more expensive (from a computational perspective) than the standard n-gram approach. Our system is currently implemented at a prototype level only, which may in practice limit its use to applications that do not require real-time language generation. Thus, as future work we intend to examine the issue of

search optimisation, and also to remove the existing noise from the training corpus. With these tasks accomplished, we expect to obtain more computationally efficient language models for practical NLG applications.

Acknowledgments The authors acknowledge financial support by FAPESP (grant nr.2009/08499-9), and are also thankful to the anonymous reviewers for their comments to improve this manuscript.

References

1. Reiter E (2007) An architecture for data-to-text systems. In: European natural language generation workshop (ENLG-2007), pp 97–104
2. Langkilde I (2000) Forest-based statistical sentence generation. In: Proceedings of ANLP-NAACL'00, pp 170–177
3. Varges S (2006) Overgeneration and ranking for spoken dialogue systems. In: Proceedings of the 4th international natural language generation conference (INLG-2006), Sydney, Australia, pp 20–22
4. Belz A (2008) Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models. Nat Lang Eng 14(4):431–455
5. DeVault D, Traum D, Arstein R (2008) Practical grammar-based NLG from examples. In: Proceedings of the 5th international natural language generation conference (INLG-2008), Columbus, USA, pp 77–85
6. Novais EM, Paraboni I (2011) Highly-inflected language generation using factored language models. In: 12th International conference on intelligent text processing and computational linguistics (CICLing-2011). LNCS, vol 6608. Springer, Berlin-Heidelberg, pp 429–438
7. Nunes MGV, Vieira FMC, Zavaglia C, Sossolote CRC, Hernandez J (1996) A construção de um léxico para o português do Brasil: lições aprendidas e perspectivas. II PROPOR, pp 61–70
8. Bilmes J, Kirchhoff K (2003) Factored language models and generalized parallel backoff. In: Proceedings of HLT-NAACL-2003, vol 2, pp 4–6
9. Muniz MCM (2004) A construção de recursos linguístico-computacionais para o português do Brasil: o projeto de Unitex-PB. Msc. dissertation, ICMC/USP
10. Maziero EG, Pardo TAS, di Felippo A, Dias-da-Silva BC (2008) A Base de Dados Lexical e a Interface Web do TeP 2.0–Thesaurus Eletrnico para o Portugus do Brasil. VI Workshop on information and human language technology (TIL-2008), pp 390–392
11. Corston-Oliver S, Gamon M, Ringger E, Moore R (2002) An overview of Amalgam: a machine-learned generation module. In: Proceedings of the international natural language generation conference (INLG-2002), pp 33–40
12. Belz A, White M, Espinosa D, Kow E, Hogan D, Stent A (2011) The first surface realisation shared task: overview and evaluation results. In: Proceedings of the 13th European workshop on natural language generation, pp 217–226
13. Belz A, Bohnet B, Mille S, Wanner L, White M (2012) The surface realisation task: recent developments and future plans. In: Proceedings of the 7th international natural language generation conference (INLG-2012), pp 136–140
14. Bohnet B, Mille S, Favre B, Wanner L (2011) StuMaBa: from deep representation to surface. In: Proceedings of the 13th European workshop on natural language generation, pp 232–235
15. Rajkumar R, Espinosa D, White M (2011) The OSU system for surface realization at generation challenges 2011. In: Proceedings of the 13th European workshop on natural language generation, pp 236–238
16. Guo Y, Hogan D, van Genabith J (2011) DCU* at generation challenges 2011 surface realisation track. In: Proceedings of the 13th European workshop on natural language generation, pp 227–229
17. Stent A (2011) ATT-0: submission to generation challenges 2011 surface realization shared task. In: Proceedings of the 13th European workshop on natural language generation, pp 230–231
18. Gervas P (2011) UCM submission to the surface realization challenge. In: Proceedings of the 13th European workshop on natural language generation, pp 239–241
19. Gatt A, Reiter E (2009) SimpleNLG: a realization engine for practical applications. In: European natural language generation workshop (ENLG-2009), pp 90–93
20. Langkilde-Geary I (2002) An empirical verification of coverage and correctness for a general-purpose sentence generator. In: Proceedings of the international natural language generation conference (INLG-2002), pp 17–24
21. Novais E, Tadeu TD, Paraboni I (2010) Improved text generation using N-gram statistics. In: 12th Ibero-American conference on artificial intelligence (IBERAMIA-2010). LNAI, vol 6433, pp 316–325. Springer, Berlin-Heidelberg
22. Novais EM, Paraboni I, da Silva Junior DFP (2012) Portuguese text generation from large corpora. In: 8th International conference on language resources and evaluation (LREC-2012), Istanbul, pp 4010–4014
23. Abreu SC, Carbonel TI, Coelho JCB, Fuchs JT, Rino LHM, Vieira R (2007) Summit: um corpus anotado com informaes discursivas visando sumarizao automtica. In: V Workshop on information and human language technology (TIL-2007), pp 1605–1610
24. Alusio SM, Specia L, Pardo TAS, Maziero E, Fortes RPM (2008) Towards Brazilian Portuguese automatic text simplification systems. The ACM Symposium on Document Engineering, pp 240–248
25. Reiter E, Sripada S (2002) Human variation and lexical choice. Comput Linguist 28(4):545–553
26. Bangalore S, Rambow O (2000) Corpus-based lexical choice in natural language generation. In: 38th Meeting of the ACL, Hong Kong, pp 464–471
27. Carpenter B (1993) Skeptical and credulous unification with applications to lexical templates and inheritance. In: Briscoe T, Copestake A, de Paiva V (eds) Default reasoning and lexical organization. Cambridge University Press, Cambridge
28. Oh A, Rudnicky A (2000) Stochastic language generation for spoken dialogue systems. In: Proceedings of the ANLP-NAACL'00 workshop on conversational systems, pp 27–32
29. Ratnaparkhi A (2000) Trainable methods for surface natural language generation. In: Proceedings of ANLP-NAACL'00, pp 194–201
30. Malouf R (2000) The order of prenominal adjectives in natural language generation. In: Proceedings of ACL-2000, Hong Kong, pp 85–92
31. Mitchell M (2009) Class-based ordering of prenominal modifiers. In: Proceedings of the 12th European workshop on natural language generation, Athens, pp 50–57
32. White M, Rajkumar R, Martin S (2007) Towards broad coverage surface realization with CCG. In: MT Summit XI workshop using corpora for natural language generation: language generation and machine translation (UCNLG+MT), pp 22–30
33. Stolcke A (2002) SRILM: an extensible language modeling toolkit. Int Conf Spoken Lang Process 2:901–904
34. da Silva Junior DFP, Paraboni I, Novais EM (2012) Um Sistema de Realização Superficial baseado em Regras para Geração de Textos em Português. USP-EACH technical report, pp 1–14
35. Papineni S, Roukos T, Ward W, Zhu W (2002) Bleu: a method for automatic evaluation of machine translation. In: ACL-2002, pp 311–318

36. NIST (2002) Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. http://www.nist.gov/speech/tests/mt/doc/ngram-study.pdf (2002)
37. Gatt A, Belz A (2010) Introducing shared tasks to NLG: the TUNA shared task evaluation challenges. In: Krahmer E, Theune M (eds) Empirical methods in natural language generation. LNAI, vol 5980, pp 264–293
38. Lucena DJ, Pereira DB, Paraboni I (2010) From semantic properties to surface text: the generation of domain object descriptions. Inteligencia Artif 14(45):48–58