Automatic Corpus Extension for Data-Driven Natural Language Generation

Elena Manishina, Bassam Jabaian, Stéphane Huet, Fabrice Lefèvre

Automatic Corpus Extension for Data-driven Natural Language Generation

Elena Manishina, Bassam Jabaian, Stéphane Huet, Fabrice Lefèvre
LIA-CERI, University of Avignon, France,
elena.manishina@alumni.univ-avignon.fr,
{bassam.jabaian,stephane.huet,fabrice.lefevre}@univ-avignon.fr

Abstract

As data-driven approaches started to make their way into the Natural Language Generation (NLG) domain, the need for automation of corpus building and extension became apparent. Corpus creation and extension in data-driven NLG domain traditionally involved manual paraphrasing performed by either a group of experts or with resort to crowd-sourcing. Building the training corpora manually is a costly enterprise which requires a lot of time and human resources. We propose to automate the process of corpus extension by integrating automatically obtained synonyms and paraphrases. Our methodology allowed us to significantly increase the size of the training corpus and its level of variability (the number of distinct tokens and specific syntactic structures). Our extension solutions are fully automatic and require only some initial validation. The human evaluation results confirm that in many cases native users favor the outputs of the model built on the extended corpus.

Keywords: corpus building, natural language generation, automatic paraphrasing

1. Introduction

As data-driven approaches started to make their way into the Natural Language Generation (NLG) domain, the need for automation of corpus building and extension became apparent. Building the training corpora manually is a costly enterprise which requires a lot of time and human resources. Such manually built corpora are often small in size and rather monotone in their content, especially in limited domain tasks. Generation systems built on such corpora show little variety in their outputs, thus making the system-user interaction less exciting for the later.

The problem of diversifying generation module outputs has recently been gaining interest within the NLG research community, mainly because the users show manifest preference for systems with varied output, which closely resembles that of a human interlocutor (Mairesse and Young, 2014).

Corpus creation and extension in data-driven NLG domain traditionally involved manual paraphrasing performed by either a group of experts or with resort to crowd-sourcing. Yet, outside NLG, automatic synonym and paraphrase extractions are well established Natural Language Processing (NLP) research fields with elaborate methodology and a large number of available tools and resources which are widely used in various NLP applications. Moreover, automatic extraction methods have been shown to perform at a level close to that of human annotators (e.g. see (Barzilay and McKeown, 2001)). In our study we propose the following extension solutions, which we combine in order to produce a richer and a more diverse training corpus:

- **Extending the system vocabulary with automatically obtained synonyms** which consists in replacing open class words (nouns, verbs, adjectives, adverbs) with their synonyms acquired automatically, thus creating new variants of existing sentences;
- **Diversifying the set of syntactic structures by introducing multi-word paraphrases** which consists in replacing sub-phrases inside a sentence with automatically obtained paraphrases;
- **Making the system responses more 'human' and user-friendly by means of introducing a modal component** which is responsible for a pragmatic/emotional facet of generated sentences.

All these solutions are fully automatic and require only initial methodology validation and assessment, performed by human experts.

This paper is structured as follows. First in Section 2. we outline the work that has been carried out previously in the area of corpus extension, as well as in synonym and paraphrase extraction domains. In Section 3. we describe the initial corpus that we used in our experiments. Section 4. introduces our methodology for corpus extension, specifically synonyms and paraphrases integration (subsections 4.1. and 4.2.) and the use of the modal component (subsection 4.3.). In Section 5. we describe our language generation paradigm. In Section 6. we present the deployment and the results of the human evaluation that we performed in order to examine the effect of our corpus extension methodology on the quality of system-user interaction. Finally, we conclude the present work with the discussion of the results and the outline of some future directions in Section 7..

2. Related Work

There have been a number of studies dealing with diversifying the output of an NLG system. All of them focus on paraphrase generation and most of them have resorted to crowd-sourcing in order to obtain paraphrases, which then required manual validation, often performed by system developers, e.g. (Mairesse and Young, 2014) or (Mitchell et al., 2014). Yet outside NLG domain automatic synonym and paraphrase extraction have been widely used in other NLP applications: machine translation (MT) and MT evaluation (Marton et al., 2009), text summarization (Barzilay et al., 1999), plagiarism detection (Sandhya and Chitrakala, 2011), etc.

Synonym and paraphrase extraction are well-established research domains with elaborate methodology and a large number of publicly available resources. As for synonym

extraction, various paradigms have been proposed and studied. They can be divided into two major classes: rule-based and data-driven. Rule-based methods generally use a knowledge base, a lexical ontology or a dictionary as a source of synonyms and semantically related words. In data-driven scenario synonyms are extracted directly from a corpus, e.g. (Van der Plas and Tiedemann, 2006). This methodology is appealing as it does not require human expertise, nor any particular ontology or database. Also various distributional similarity approaches have been explored (Freitag et al., 2005).

As for paraphrases, a variety of approaches have been tested including extracting paraphrases from multiple translations of the same source text (Barzilay and McKeown, 2001) (parallel corpus approach), using monolingual comparable corpora (Wang and Callison-Burch, 2011), aligning several dictionary definitions of the same term to extract paraphrases (Murata et al., 2005), etc.

Concerning the modal component, to our knowledge, there have been no studies that investigate extending the NLG model with an emotional component having a specific lexical realization. This problem is usually treated from an Artificial Intelligence perspective and not a linguistic one; thus, sentiment transmission is reduced to mimics, gestures and voice pitch modifications (prosody). In our study we cast the problem of modality as a linguistic or rather an NLP problem, as sentiments may have a lexical and in part a syntactic realization.

3. Building a Basic Corpus from Existing Resources

To create an initial corpus for our study we used the data collected for the TownInfo project (Young et al., 2010). This project uses a template-based generation paradigm, where templates represent the mappings between given semantic realizations and model output sentences in a natural language.

In the semantic formalism used in the TownInfo project each dialog act is represented by a dialog act type (a so-called semantic frame: *inform*, *negate*, etc.) and a set of semantic concepts in the form of key-value pairs. We employed the set of templates used in the template-based NLG module and the database of possible values (see Figure 1) to build a basic training corpus with the goal to test our extension methodology on this corpus later on. Thus the training sentences were created from the templates by replacing the variables (names, dates and numbers) with the values from the database.

DA template:
inform(name=X, type=Y, area=Z, food=W)
$\Rightarrow$ *X is a Y in Z serving W food.*
$\otimes$
Values: ***bar_metropol, bar, north, french***
$\Downarrow$
Utterance: ***Bar Metropol*** *is a* ***bar*** *in the* ***north*** *part of town serving* ***French*** *food.*

Figure 1: Dialog act example.

The corpus obtained that way is relatively small (several thousand sentences) and it consists of repetitive uniform sentences with very limited vocabulary and low variability in syntactic structures (see Table 1 for detailed statistics). At the same time it represents a solid base for further extension.

General	
Size in words	34283
Size in sentences	3151
Number of tokens	486
Tokens	
- nouns	219
- verbs	109
- adjectives	80
- adverbs	58

Table 1: Corpus statistics before extension.

4. Extension Solutions: Methodology

4.1. Integrating Synonyms

We start off by creating new sentences via integrating synonyms. Synonyms are words that denote the same semantic concept and are interchangeable in different contexts[1]:

Ref.: They **serve** Italian food.
Syn1: They **offer** Italian food.

We aim at creating new sentences for the training corpus by means of replacing open class (non-functional) words with their synonyms in existing ones. In general, phrases where words are replaced with their synonyms can be regarded as paraphrases, for example:

It is a **nice** bar **serving** Russian food
It is a **lovely** bar **offering** Russian food

Such synonym-based paraphrases are easier to obtain than large-scale multi-word paraphrases and they have less possibility of being grammatically incorrect or dis-fluent. For our task we adopt two different methods for synonym extraction: an ontology-based method and the approach based on word-to-word parallel corpus alignment described in (Bannard and Callison-Burch, 2005). Ontology-based methods remain the most reliable, as they consist in retrieving synonym which have already been designated as such by human experts. Corpus-based synonym extraction method consists in retrieving aligned pairs of words from a bi-directional word-aligned parallel corpus. Matching pairs extracted from both target-source and source-target directions are considered to be synonyms.

Corpus-based methods (like the one used by Bannard and Callison-Burch) despite their appeal as 'fully autonomous' do not achieve the precision of ontology-based methods, though the recall is much higher. At the same time ontologies are often limited to a specific domain and a particular language. In order to balance precision and recall we combine an ontology-based and a corpus-based methods.

[1] Linguistics Glossary: `www.lingualinks.com`

Ref:	*OK , 10 pounds per* ***person***	*and they play* ***ethnic*** *music*
Syn1:	*OK , 10 pounds per* ***man***	*and they play* ***folk*** *music*
Syn2:	*OK , 10 pounds per* ***human***	*and they play* ***indigenous*** *music*
Syn3:	*OK , 10 pounds per* ***individuum***	*and they play* ***national*** *music*
Syn4:	*OK , 10 pounds per* ***head***	*and they play* ***local*** *music*
Ref:	*OK a hotel in any* ***price*** *range*	*Chez Sergu serves Chinese* ***food***
Syn1:	*OK a hotel in any* ***monetary value*** *range*	*Chez Sergu serves Chinese* ***cuisine***
Syn2:	*OK a hotel in any* ***cost*** *range*	*Chez Sergu serves Chinese* ***nutrient***
Syn3.	*OK a hotel in any* ***value*** *range*	*Chez Sergu serves Chinese* ***dishes***
Syn4:		*Chez Sergu serves Chinese* ***meal***

Table 2: New training examples built with synonyms (before filtering).

4.1.1. Extracting Synonyms from Wordnet

For the ontology-based method we used a publicly available Wordnet ontology (Fellbaum, 1998). Wordnet is very well suited for our task as its structure allows for a precise and controllable synonym identification.

Words in Wordnet are grouped into sets of cognitive synonyms (synsets); synset is an abstract representation of a specific meaning of the concept; each synset is represented by a set of lemmas, i.e. words describing a specific meaning of a given concept.

We start with selecting words in our corpus to be replaced with potential synonyms. The candidate words must belong to one of the open grammatical classes (common nouns, verbs, adjectives and adverbs). Each selected word is then regarded as a centroid concept in Wordnet. We extract all lemma names from the first two synonym sets of this concept and replace them for corresponding words in the initial sentence. The number of synsets to be considered was selected based on the assumption that the direct (non-metaphoric) meanings for each concept are presented before the metaphoric ones; we are mostly interested in the direct meanings.

4.1.2. Extracting Synonyms from the Corpus

The second method requires a substantial general-domain parallel corpus aligned at the word level. We used the French-English version of the Europarl corpus (Koehn, 2005) mainly because of its size, but the corpus pair is not significant as long as one of the languages is the language of the initial NLG corpus. For each designated word, we extract all corresponding translations from the aligned corpus. Then for each distinct target equivalent of a given word, we look for all the aligned counterparts in the source, thus getting different translation of the same target word in the source which we consider to be synonyms. We also keep the number of occurrences for each translation pair; pairs with the number of occurrences less than 2 are dropped.

4.1.3. Filtering the Synonyms

Obviously, not all word pairs obtained the way described in the previous section are valid synonyms (see Table 2). We implement a two-step automatic filtering which is intended to remove, first, non-valid synonym pairs, and second, if some of those pairs made their way into the corpus, to remove incoherent sentences. The newly constructed sentences which passed both filtering steps are added to the final corpus. In practice filtering consists in, first, pruning of the newly created synonym dictionary based on the distributional semantics (DS) similarity scoring, and secondly by calculating n-gram language model (LM) score for each new sentence.

To implement a DS-filtering we generate word vectors for extracted synonyms and their reference words. The word2vec model (Mikolov et al., 2013) is trained on LDC Gigaword 5th edition, the Brown corpus and the English Wikipedia using skip-gram algorithm with a vector size of 300 and a 10-word window. We compare the vectors by calculating pairwise similarities between the reference and synonyms vectors for each set of synonyms. For each set of vectors a threshold is determined by taking the average similarity score for each set. Then we keep the words with vectors sufficiently close to the reference vector and remove the others.

LM-filtering method uses a 5-gram language model built on the Web corpus. N-gram language models are widely used for estimating fluency and validating the output of various NLP systems. It is particularly important for tasks involving generation of phrases in a natural language, such as machine translation and natural language generation. Thus, we calculate an LM score for each newly produced sentence and eliminate the ones with a low score.

4.2. Integrating Paraphrases

The second step in our corpus extension pipeline is automatic extraction and integration of paraphrases for selected contiguous sequences of words. Paraphrases are reproductions of the same meaning having different surface forms. Paraphrases can be viewed as multi-word synonyms as they are interchangeable in the same context. For example:

Do you like Italian or Chinese food?
Would you prefer Italian or Chinese food?
Would you rather have Italian or Chinese food?

Our goal is to create new training instances by means of replacing selected subphrases with their paraphrases in existing corpus sentences.

Previous research on paraphrasing for NLG mainly focused on entire utterances, thus making it harder to extract auto-

⊗modal(*There is no restaurant matching your request*)

⇓

***I am sorry**, there is no restaurant matching your request.*

⊗modal(*Do you want Italian or Chinese food?*)

⇓

*Do you want Italian or Chinese food, **I wonder**?*

Figure 2: Modal component examples.

matic paraphrases, long sentences being scarce and therefore hard to find in the corpus. In our study we focus on paraphrasing sub-phrases: smaller and more frequently occurring units. We split each sentence into contiguous chunks of different lengths, thus augmenting the odds to find paraphrases for these chunks. This method cannot be used to identify large-scale paraphrases containing gaps. But in our case it is not critical as all phrases to be paraphrased are themselves contiguous chunks of the maximum length of 4 words.

We used the same word-aligned parallel corpus as in the previous section for isolated words. However this time we retrieved contiguous chunks of words consistent with the alignment.

Only the blocks of words which do not contain variables (proper nouns, like hotel names, addresses, etc.) were regarded as sources for potential paraphrases. Each multiword expression was further split into smaller units to get phrases of different lengths in order to extend the coverage. There are several conditions that we apply:

1. minimum length of a subphrase is 2
2. at least one word should be an open-class word (noun, verb, adjective, adverb) with a single exception of modal verbs (can, may, etc.)
3. subphrases are contiguous

To build new corpus instances we replace subphrases with their paraphrases within the original sentence. Then we run the same LM-filtering as in case of sentences with synonyms to remove incoherent sentences.

4.3. The Modal Component

The last extension component is adding the so-called modal expressions. According to the Linguistics Glossary, modality is a facet of illocutionary force, signaled by grammatical or lexical devices, that expresses the illocutionary point or general intent of a speaker.

Modal clauses do not change the semantic content of generated phrases in any way and thus can be securely added and removed from the output (see Fig. 2). They are not supposed to interfere with the users' understanding of the phrase, but rather create a more friendly atmosphere during communication. Also in our setup modals are always added in a form of an adjunct. Thus they do not affect in any way the grammatical structure and coherence of the utterance.

There are 14 semantic frames in our corpus: *inform, request, negate, select, hello, confirm, reqmore, repeat, ack, affirm , bye, deny, thankyou, reqalts*. Each frame represents a particular discourse type with a well-defined semantic content. Each of these frames may take a given set of modal expressions to augment its elocutionary force.

We incorporated a modal in a form of an additional concept to each input sequence of concepts. We started with manually crafting one generic modal per discourse type; then we apply the technique for paraphrase acquisition described in the previous section in order to obtain paraphrases for each modal.

5. NLG Model

In our work we adopt the approach where the task of NLG can be regarded as the translation between a formal meaning representation and a natural language, and therefore, can be performed using statistical machine translation techniques. A similar approach has already been applied to semantic interpretation, e.g. in (Jabaian et al., 2013; Jabaian et al., 2016).

Our generation pipeline allows for inclusion and combination of different generation models (in our case an n-gram model and Conditional Random Fields (CRFs)-based model) and uses an efficient decoding framework (finite-state transducers' best path search).

Unlike (Oh and Rudnicky, 2000) which used n-gram modeling in its pure form to build a natural language generator, we would like to extend the n-gram generation model, adding a reordering scheme and a target LM, in accordance with a standard statistical machine translation pipeline. We follow the approach introduced in (Lavergne et al., 2011) and build a framework which represents a composition of finite-state transducers as follows:

$$e^* = \text{bestpath}(r(conc) \circ tm \circ lm) \tag{1}$$

where *r(conc)* is the graph containing the reordered source concepts, *tm* is the translation model and *lm* is a target language model. The reordering model represents the reordering rules learned from the training corpus.

In the initial configuration, *tm* in an n-gram model defines the joint probabilities of a given set of concepts and their lexical realization $\Pr(conc, lex)$. The probability of a given sequence of tuples is expressed as a standard n-gram model probability:

$$\Pr(conc, lex) = \prod_{k=1}^{K} \Pr((conc, lex)_k | (conc, lex)_{k-1}, ..., (conc, lex)_{k-n+1}) \tag{2}$$

The target language model *lm* is an n-gram language model which ensures the fluency and the overall grammatical correctness of the output:

$$\Pr(lex_1 ... lex_I) = \prod_{i=1}^{I} \Pr(lex_i | lex_{i-n+1} ... lex_{i-1}) \tag{3}$$

The decoding is performed on a joint Finite State Transducer (FST) graph; the role of the decoder is to find the

General	
Size in words	261085
Size in sentences	18755
Number of tokens	598
Tokens	
- nouns	276
- verbs	165
- adjectives	107
- adverbs	65

Table 3: New corpus statistics (after extension).

highest scoring path in the graph which is supposed to be the best generation hypothesis according to the model. The interest of this approach is that it allows to combine a number of modules in a pipeline and to progressively refine a solution for the translation task at hand, see for instance (Jabaian and Lefèvre, 2013) for an application of this methodology to error correction of speech recognizer outputs.

A discriminative version of *tm* in the initial combination of transducers (presented in Equation 1) models the conditional probability of the lexical side given a set of concepts, i.e. $\Pr(lex|conc)$. The conditional probability of the lexical realization given a concept can be modeled with CRFs. Thus the discriminative pipeline is obtained by replacing the original n-gram model with the CRF-based one.

Given matched sequences of observations x_1^L and labels y_1^L CRFs expresses the conditional probability of labels as:

$$\Pr(x_1^L, y_1^L) = \frac{1}{Z(x_1^L; \theta)} \exp(\theta_T G(x_1^L, y_1^L)) \qquad (4)$$

The joint version of the pipeline, which integrates both n-gram and CRFs, represents a composition of transducers which can be expressed as follows:

$$e^* = \text{bestpath}(r(conc) \circ tm_1 \circ tm_2 \circ lm) \qquad (5)$$

where tm_1 is an n-gram model graph and tm_2 is a graph containing CRF scores. This generation model is described in more details in (Manishina, 2016).

6. Experiments and Human Evaluation

The integration of synonyms more than tripled the size of the training corpus: 11k sentences versus 3k in the original corpus. Further extension with paraphrases yielded additional 7.5k phrases raising the number of sentences in the final corpus to 18.7k and the number of unique tokens to 598 (Tables 3 and 4).

In order to validate our extension methodology we perform a human evaluation. Our goal is to examine the effect of corpus extension on the quality of user-system interaction and on the users' perception of system responses, specifically the responses produced by the system trained on the extended corpus. We split the participants (56) into three user groups — native speakers of English and non-native speakers. The non-native speakers group is further divided into an advanced and an intermediate level groups.

As shown in Figure 3 we present the users with 3 variants of the same sentence: one produced by a template-based generator (referred as *template* in Table 5), one produced by the NLG statistical model (presented in Section 5.) trained on the initial corpus (*initial_c* in Table 5), and finally, one produced by the same statistical model trained on the extended corpus (*extended_c* in Table 5). The users were asked to score each sentence according to 3 criteria: fluency, informativeness and naturalness/diversity, the last one being the targeted score for assessment of our corpus extension methodology. We also asked the users to rank the three sentences according to their personal preferences. The results are presented in Table 5.

Interestingly, according to the native users' assessments, the extended system is comparable in fluency to the template-based generator (4.15 vs 4.19) and is equivalent to the later in terms of naturalness, being at the same time slightly higher in informativeness. This is a noteworthy phenomenon: possibly, the difference in vocabulary might affect the perception of the overall informational content of the phrase for a native speaker. This contrasts with what we observed in the assessments of the non-native speakers, who sometimes dismiss extended system output as less natural and less fluent.

In general non-native speakers favor template-based outputs which use simple common vocabulary and structures. At the same time the equal distribution of the extended system rankings and 35% of phrases ranked as best outputs in the 'advanced group' might suggest that we may be dealing with occasional vocabulary 'strangeness' and not a systematic phenomenon.

On the contrary, native speakers give equal scores to templates and extended system output in terms of naturalness and fluency (due to their familiarity with non-common vocabulary). For example, comparing the outputs of template-based (a) and extended (b) systems:

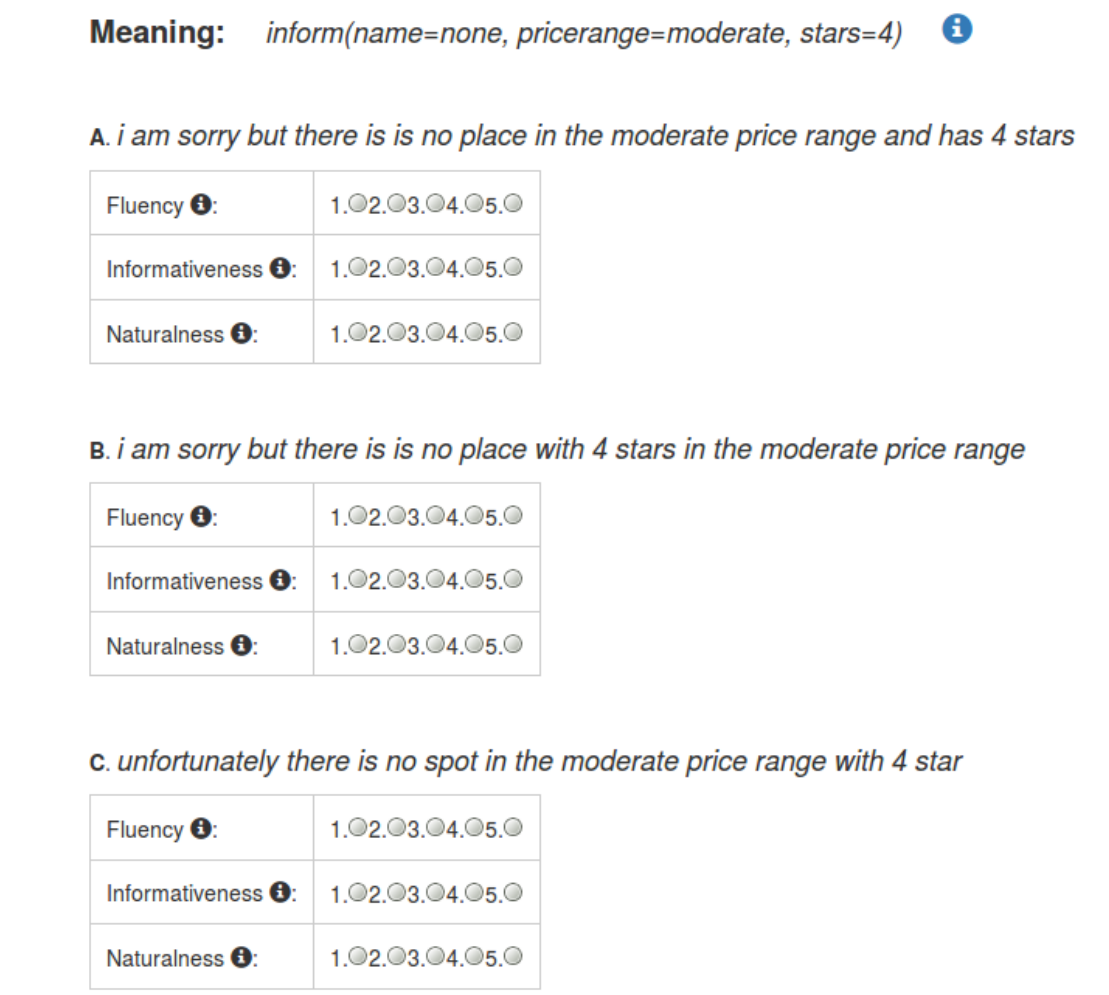
*(a) Art House hotel is a **restaurant** on Art Square serving English food near **cinema**.*
*(b) Art House hotel is an **eating place** on Art Square serving English dishes not far from **movie house**.*

non-native speakers show a manifest preference for the output (a) provided by the template-based system, while the native speakers score the output (b) considerably higher. We also noticed that in many cases the overall output perception affects all criteria at once in the non-native groups: if the output does not seem natural, it is almost automatically scored as less fluent and less informative.

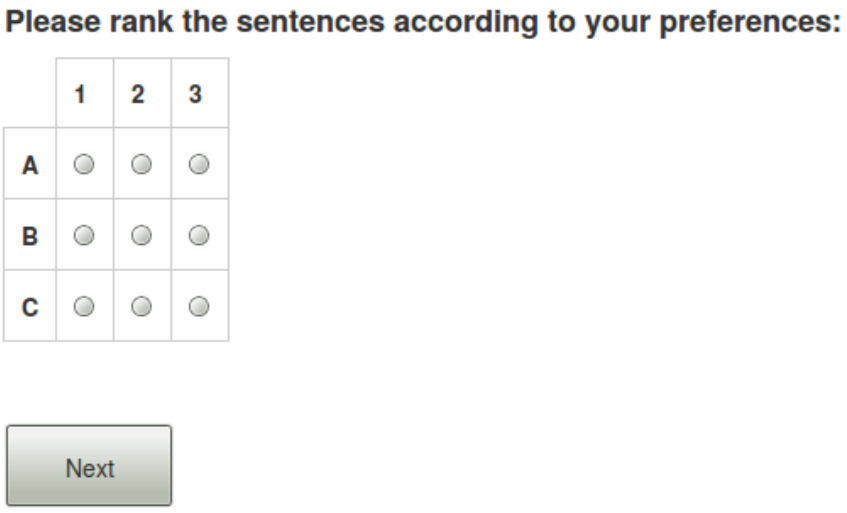
The above discussion leads us to the conclusion that, different user backgrounds shape differences in judgments. This raises a question: should an NLG module be oriented towards a particular user group or should it be generic enough to target as many users as possible? Obviously it depends on the task at hand, the context and the general purpose of the system. International dialog systems, targeting non-native speakers, might consider employing basic simple phrases which use standard English vocabulary, with no slang or local variations (like 'movie house' in the example above).

7. Conclusion

By means of synonym and paraphrase extraction techniques and modality insertions we have created an extended and



(a)



(b)

Figure 3: Evaluation interface: (a) scoring and (b) ranking.

diverse NLG training corpus which can be used not only for training other NLG models, but also in language understanding and other NLP domains. Apart from the modal component, which is language- and task-specific, all the steps in corpus extension are automatic and do not require any human intervention (except for the evaluation); they can be used to enrich the training corpus in any domain and any language.

Integration of synonyms into the training corpus is a simple, yet an effective solution which has not been fully explored

Extension procedure	Number of sentences	Number of tokens
Initial corpus	3151	486
Extended corpus (synonyms)	11175	589
Extended corpus (synonyms+paraphrases)	18755	598

Table 4: Corpus statistics after extension: synonyms and paraphrases.

and applied to NLG up to now. Paraphrasing is a well-known technique largely used in NLG domain for corpus extension (usually with resort to crowd-sourcing or manual paraphrasing), yet it is also very effective and less expensive when implemented as an automatic process which does not require human intervention.

Paraphrasing parts of a phrase, and not the entire phrase, creates smaller reusable subphrases, thus allowing for different combinations of subphrases, which in turn adds more variability and further augments the size of the training corpus. As human evaluation showed, training a statistical NLG model on the extended corpus generates outputs that are more diverse and appealing to native speakers. Nevertheless the scope of the system (in terms of end-user coverage) should also be taken into consideration when building the training corpus, specifically when this scope includes non-native speakers.

8. Acknowledgments

This work is partially funded by the ANR MaRDi project (ANR CONTINT 2012 ANR-12-CORD-0021).

9. Bibliographical References

Bannard, C. and Callison-Burch, C. (2005). Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 597–604. Association for Computational Linguistics.

Barzilay, R. and McKeown, K. R. (2001). Extracting paraphrases from a parallel corpus. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 50–57. Association for Computational Linguistics.

Barzilay, R., McKeown, K. R., and Elhadad, M. (1999). Information fusion in the context of multi-document summarization. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 550–557. Association for Computational Linguistics.

Fellbaum, C. (1998). *WordNet*. Wiley Online Library.

Freitag, D., Blume, M., Byrnes, J., Chow, E., Kapadia, S., Rohwer, R., and Wang, Z. (2005). New experiments in distributional representations of synonymy. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*, pages 25–32. Association for Computational Linguistics.

Jabaian, B. and Lefèvre, F. (2013). Error-corrective discriminative joint decoding of automatic spoken language transcription and understanding. In *INTERSPEECH*, pages 2718–2722.

Jabaian, B., Besacier, L., and Lefèvre, F. (2013). Comparison and combination of lightly supervised approaches for language portability of a spoken language understanding system. *Audio, Speech, and Language Processing, IEEE Transactions on*, 21(3):636–648.

Jabaian, B., Lefèvre, F., and Besacier, L. (2016). A unified framework for translation and understanding allowing discriminative joint decoding for multilingual speech semantic interpretation. *Computer Speech & Language*, 35:185–199.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *MT Summit*.

Lavergne, T., Crego, J. M., Allauzen, A., and Yvon, F. (2011). From n-gram-based to crf-based translation models. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 542–553. Association for Computational Linguistics.

Mairesse, F. and Young, S. (2014). Stochastic language generation in dialogue using factored language models. *Computational Linguistics*, 40(4):763–799.

Manishina, E. (2016). *Data-driven Natural Language Generation Using Statistical Machine Translation and Discriminative Learning*. Ph.D. thesis, University of Avignon.

Marton, Y., Callison-Burch, C., and Resnik, P. (2009). Improved statistical machine translation using monolingually-derived paraphrases. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 381–390. Association for Computational Linguistics.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Mitchell, M., Redmond, W., Bohus, D., and Kamar, E. (2014). Crowdsourcing language generation templates for dialogue systems. *Proceedings of the International Conference on Natural Language Generation and the Special Interest Group on Discourse and Dialogue*, page 16.

Murata, M., Kanamaru, T., and Isahara, H. (2005). Automatic synonym acquisition based on matching of definition sentences in multiple dictionaries. In *Computational Linguistics and Intelligent Text Processing*, pages 293–304. Springer.

Oh, A. H. and Rudnicky, A. I. (2000). Stochastic language generation for spoken dialogue systems. In *Proceedings of the 2000 ANLP/NAACL Workshop on Conversational systems-Volume 3*, pages 27–32. Association for Computational Linguistics.

Sandhya, S. and Chitrakala, S. (2011). Plagiarism detection of paraphrases in text documents with document

Users	**Native**			**Advanced**			**Intermediate**		
Model	**init_c**	**extended_c**	**template**	**init_c**	**extended_c**	**template**	**init_c**	**extended_c**	**template**
Fluency	4.1	4.2	4.2	4.2	3.6	4.3	4.2	3.6	4.4
Informativeness	4.6	4.4	4.4	4.5	4.2	3.4	4.5	4.0	4.5
Naturalness	3.9	4.0	4.0	4.1	3.4	4.1	4.1	3.3	4.3
Rank 1	50%	42%	42%	55%	35%	59%	44%	29%	51%

Table 5: Human evaluation results.

retrieval. In *Advances in Computing and Information Technology*, pages 330–338. Springer.

Van der Plas, L. and Tiedemann, J. (2006). Finding synonyms using automatic word alignment and measures of distributional similarity. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 866–873. Association for Computational Linguistics.

Wang, R. and Callison-Burch, C. (2011). Paraphrase fragment extraction from monolingual comparable corpora. In *Proceedings of the 4th Workshop on Building and Using Comparable Corpora: Comparable Corpora and the Web*, pages 52–60. Association for Computational Linguistics.

Young, S., Gašić, M., Keizer, S., Mairesse, F., Schatzmann, J., Thomson, B., and Yu, K. (2010). The hidden information state model: A practical framework for pomdp-based spoken dialogue management. *Computer Speech and Language*, 24(2):150–174.