Statistics-Based Lexical Choice for NLG from Quantitative Information

Xiao Li and **Kees van Deemter** and **Chenghua Lin**
Computing Science department
University of Aberdeen
King's College
Aberdeen, AB24 3FX, UK
{xiao.li, k.vdeemter, chenghua.lin}@abdn.ac.uk

Abstract

We discuss a fully statistical approach to the expression of quantitative information in English. We outline the approach, focussing on the problem of Lexical Choice. An initial evaluation experiment suggests that it is worth investigating the method further.

1 Introduction

NLG systems express information in human language. To do this well, these systems need to "know"what expressions are most suitable for expressing a given piece of information. The most direct way to define words in NLG systems is manual coding, as it was done in systems such as FoG (Golberg et al., 1994) and SumTime-Mousam (Sripada et al., 2003). However, manual coding is time consuming, it can be argued to be theoretically unsatisfactory, and it is error prone even when performed by domain experts. The process is complicated in the fact that words like *pink* (Roy, 2002) and *evening* (Reiter et al., 2005) have different meanings for individual speakers.

Recent NLG approaches learn the use of words through statistical analysis of data-text corpora. For example, Belz's semi-automatic system for weather forecasting automatically learns a grammar based on a pre-existing (i.e., manually coded) set of grammar rules (Belz, 2008). Liang et al. (2009) developed a fully statistical alignment-based algorithm that automatically acquires a mapping from quantitative information to English words by adopting a hierarchical hidden semi-Markov model trained by Expectation Maximization. Konstas and Lapata (2013) introduced a generation model based on Liang's algorithm . However, these existing approaches have difficulty handling situations in which a word expresses a *combination* of data dimensions, for example as when the word "mildëxpresses a combination of warm temperatures and low wind speed.

In this paper, we discuss a new approach to the problem; the approach is fully statistical and it is able to handle situations in which a word or phrase maps to a combination of data dimensions. We focus on Lexical Choice but are investigating applications to other areas of NLG.

2 Methodology

In many areas of perception research, a method called "contour stylizationïs employed to mimic a complex signal (i.e., a complex graph) by means of a limited number of straight lines (Johan t Hart and Cohen, 1990). Our method uses the similar idea and applies it to two dimensions (i.e., weather data and language) at the same time. Our approach builds a bridge between quantitative information and words by discretising the data.

2.1 Representing Data in Vector

A continuous dimension can be represented by a set of discrete parameters, so called **key-points**. For example, wind speed (ws) is a continuous dimension with its value between 0 knot to 36 knots. A group of key-points can then be used to represent any value of wind speed. For instance, a possible key-point group is $\{ws = 0, ws = 12, ws = 24, ws = 36\}$, in which key-points are evenly spaced. The aim of using key-points is to transform the original quantitative dimension into probability dimensions. This process is similar to Signal Analysis (Reiter 2007)

Proceedings of The 9th International Natural Language Generation conference, pages 104–108,
Edinburgh, UK, September 5-8 2016. ©2016 Association for Computational Linguistics

in which each key-point plays a role as a Signal Sensor. In the above example, 5 key-points are used to represent wind speed collectively, where each key-point specifies a specific range of wind speed. In this way, if a word describes wind speed within a certain range, we will find the connection of the word to the relative key-points.

Based on this formulation, any wind speed can be represented by weighted key-points through linear interpolation. Suppose one would like to represent an arbitrary wind speed, say $ws = 5$. Note that $ws = 5$ falls between the range of key-points $ws = 0$ and $ws = 12$ as described above. Using linear interpolation, one can derive the weights of key-points $ws = 0$ and $ws = 12$ for representing $ws = 5$, which are 0.58 and 0.42 respectively. Because the remaining key points does not contribute to represent wind speed $ws = 5$, their weights are set to 0. Finally, the wind speed $ws = 5$ can be represented as a vector $\langle 0.58, 0.42, 0, 0\rangle$, which encodes the weights for the key-point group.

Although in the above example key-points $\{ws = 0, ws = 12, ws = 24, ws = 36\}$ are set evenly spaced, it should be noted that the setting of key-points (e.g., the choice of key-point values) has relatively little impact on predicting the use of words. This is because the our method can be regarded as fitting the occurrence function of words by a straight line in the contour stylization angle (in addition to the Signal Analysis), and the key-points present the inflection points' abscissa of the straight line. Although carefully selecting key-points can possibly enhance the model's performance, our model adopt the evenly spaced key-points, which empirically works well enough in general.

2.2 Representing Words in Vector

Expressions such as words can be represented by key-points weight vectors as well. For example, in English the expression *calm* is only used to describe wind speed close to 0. So, *calm* can also be represented using the same key-point group as before, i.e., represented with a high weight for $ws = 0$ (such as 0.9, for instance), and a low weight for $ws = 12$ (e.g., 0.01). For the moment, the weights of *calm* are estimated by hand. In section 2.4 we will see how the weights can be estimated from a data-text corpus.

2.3 Lexical Choice

This section introduces how our proposed appro-cah handles the lexical choice in the NLG process through Cosine similarity. Suppose both quantitative information and lexical expressions have been converted into vectors (i.e., $\vec{q}$ and $\vec{e}$) in the same vector space parameterised by the key-points. The problem of finding the most likely expression ($\vec{e}$) for the given quantitative information ($\vec{q}$) can be transformed to the process of finding the most similar lexical expression vector $\vec{e}$ to $\vec{q}$. We exemplify the lexical choice process below, using wind speed as quantitative dimension.

Suppose the key-points are still $\{ws = 0, ws = 12, ws = 24, ws = 36\}$. The candidate expression words are *calm* and *breeze*, which can be represented in a form of key-point weight vectors as below:

$$\vec{e}_{calm} = \langle 0.9, 0.01, -0.9, -1\rangle \quad (1)$$

$$\vec{e}_{breeze} = \langle 0.7, 0.9, -0.8, -1\rangle \quad (2)$$

Now our goal is to choose the most suitable word to describe wind speed $ws = 5$ from the available candidate word expressions (i.e., *calm* and *breeze*). As discussed in Section 2.1, $ws = 5$ can also be represented by a key-point weight vector

$$\vec{q}_{ws=5} = \langle 0.58, 0.42, 0, 0\rangle \quad (3)$$

Based on the same key-point vector space, we calculate the Cosine similarities between each candidate word and the target wind speed $ws = 5$, and the most suitable word is naturally the one with the highest similarity to $ws = 5$.

$$\text{Sim}(\vec{e}_{calm}, \vec{q}_{ws=5}) = \frac{\vec{e}_{calm} \cdot \vec{q}_{ws=5}}{\|\vec{e}_{calm}\| \, \|\vec{q}_{ws=5}\|} = 0.45 \quad (4)$$

$$\text{Sim}(\vec{e}_{breeze}, \vec{q}_{ws=5}) = \frac{\vec{e}_{breeze} \cdot \vec{q}_{ws=5}}{\|\vec{e}_{breeze}\| \, \|\vec{q}_{ws=5}\|} = 0.64 \quad (5)$$

As can be seen above, the similarity between $\vec{q}_{ws=5}$ and $\vec{e}_{breeze}$ is higher than that of $\vec{e}_{clam}$. Therefore, *breeze* is a better choice for expressing $ws = 5$.

2.4 Estimating Weight Vector for Word Expressions

One key challenge in applying our approach for learning the relationship between quantitative information and words is to find the optimal vector $\vec{e}$ for each possible expression word. Suppose we have r data to text pairs denoted as $< data_i, text_i >_{i=1}^r$, where $data_i$ in the pairs consists of quantitative dimensions and $text_i$ refers to the expression words as shown in Eq. 6.

$$< data, text > \Rightarrow \{dim_{1,...,m}, exp_{1,...,n}\} \qquad (6)$$

Following section 2.1, for each data to text pair, we firstly discretise the data dimensions ($dim_{1,...,m}$) into a key-point group $\{\vec{d_1}, \vec{d_2}, ..., \vec{d_m}\} \equiv \vec{\mathbf{d}}$. Next, we can find the optimal values for weight vector $\vec{e_i}$ by solving Eq. 7 constructed based on the training data $< data_i, text_i >_{i=1}^r$.

$$\begin{bmatrix} \vec{\mathbf{d}}_1 \\ \vec{\mathbf{d}}_2 \\ \vdots \\ \vec{\mathbf{d}}_r \end{bmatrix} \vec{e_i}^T = \begin{bmatrix} \text{isOccur}(exp_i|text_1) \\ \text{isOccur}(exp_i|text_2) \\ \vdots \\ \text{isOccur}(exp_i|text_r) \end{bmatrix} \qquad (7)$$

The function isOccur$(exp_i|text_i)$ returns 1 if exp_i occurs in the corresponding $text_i$, and returns 0 otherwise.

Generally, there are fewer free parameters than the number of equations, so we can always find the optimised solution for estimating $\vec{e_i}$ using Least Square. If there are more than one solution, we adopt the solution with the least norm. In the same way, we can obtain weight vectors for all the candidate expressions.

So far we have described how to estimate the key-point weight vector for every candidate expression from training data, i.e., data-text pairs. In the test phase, to predict the most likely words for unseen data, we firstly represent data as a weight vector, and then compare its cosine similarity against every candidate expression. Since the weight vectors for expressions $\vec{e_i}$ are trained through the occurrence function isOccur(), the similarity between unseen data and a candidate expression reflects the suitability of an expression in expressing the data.

2.5 Discussion: Handling multiple dimensions

One of the important features of our approach is the ability of choosing expressions for data with multiple dimensions. We stress that both the training process and lexical choice process are applicable to multiple data dimensions. First, in the training process, information of different quantitative dimensions is converted into key-point weights, so the boundaries between different dimensions have disappeared. The training process could even calculate the implicit relationship between expressions and quantitative data. Second, the lexical choice process selects expressions based on a set of dimensions rather than each single dimension. This is why this approach can handle the multiple dimension information.

3 Evaluating the proposed approach to Lexical Choice

To perform an initial sanity check on our approach, we built a small corpus from SumTime-Meteo Corpus (Sripada et al., 2002), which contains human writing weather forecasts with meteorological data. We selected 144 wind speed forecasts with data whose wind speeds do not change a lot during a forecast period, and summarize these data into three dimensions, as shown in Table 1.

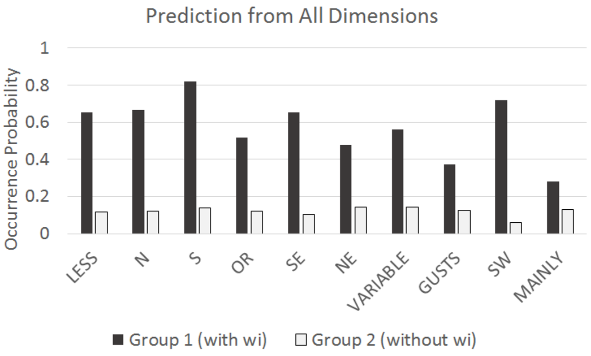
We randomly selected 96 records of the total 114 data records to train the model, and adopt the rest of data records to evaluate. We evaluated 10 words[1]: *LESS, N, S, OR, SE, NE, VARIABLE, GUSTS, WS, MAINLY*, which are the words occurring more than 5 times in the small corpus. For each candidate word w_i, we separate the testing data into two groups. Forecast texts in group 1 contain word w_i but not in group 2. When we use our model (trained with the SumTime-Meteo Corpus) to predict the occurring probability of w_i in group 1 and group 2 respectively, we expect to obtain higher occurring probability $p(w_i|G_1)$ from group 1 than $p(w_i|G_2)$ from group 2. The results are shown in Figure 1.

As shown in Figure 2, it is clear that experimental results are inline with our expectation: our approach does produce higher occurring probabilities in group 1 than in group 2. Recall that one key feature of our approach is its capability to model multiple dimensional features. To show the benefit of this feature, we have also applied our approach modelling taking into account each single dimension separately. By comparing Table 1 and Table 2, we can see that the

[1] "Wordsïn the SumTime-Meteo Corpus include abbreviations such as SW (South-West) etc., see Table 1 for examples of text fragments and data.

Tabel 1: Some sample records of our corpus.

	Wind Speed	Wind Direction	Wind Variance
MAINLY W-NW 10 OR LESS	4.2	282	7
VARIABLE 8 OR LESS	7.5	319	12
...	...	...	...



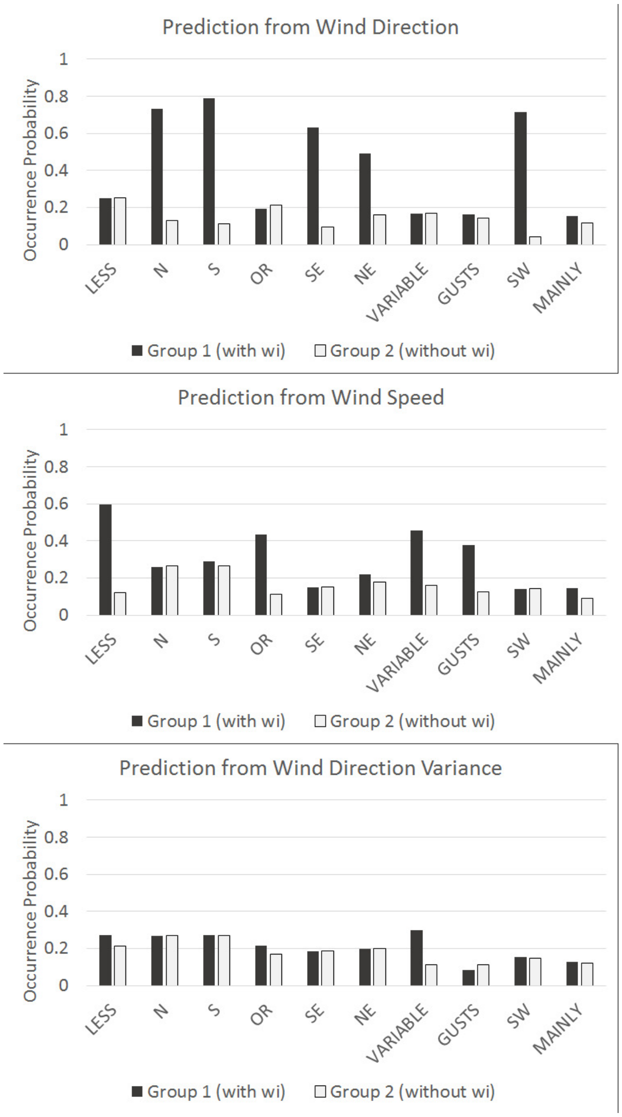
Figuur 1: The predicted occurring probabilities based on data of all dimensions.

prediction performance of words based on multiple dimension outperforms all the models considering a single dimension only, especially when predicting words *variable* and *mainly*.

4 Conclusion

We have sketched an approach to choosing lexical expressions according to multiple quantitative information. To have this ability, this approach learns the relationship between quantitative information and words by the following steps: a) resolving quantitative information and the occurrence of expressions into the same linear space; b) building equations of expressions' weight vector; c) finding the best solution of the equations. Initial evaluation suggest that this approach may be on the right track.

The possibility of applications to Lexical Choice in Natural Language Generation is perhaps most obvious, but the mapping that we learn is applicable to interpretation as well. In other words, our proposal aims to solve the age-old problem in Linguistics and Fuzzy Logic of how to specify the meaning of vague words (Van Deemter, 2012), which resists traditional approaches to semantics, because these words admit borderline cases.



Figuur 2: The predicted occurring probabilities based on data of single dimension: *wind direction*, *wind speed*, *and wind direction variation*.

Acknowledgments

This work is supported by the award made by UK Economic & Social Research Council (ESRC); award reference ES/M001628/1.

References

Anja Belz. 2008. Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models. *Natural Language Engineering*, 14(04):431–455.

Eli Golberg, Richard Kittredge, and Norbert Driedger. 1994. A new approach to the synthesis of weather forecast text. *IEEE Expert*.

Rene Collier Johan t Hart and Antonie Cohen. 1990. A perceptual study of intonation.

Ioannis Konstas and Mirella Lapata. 2013. A global model for concept-to-text generation. *J. Artif. Intell. Res.(JAIR)*, 48:305–346.

Percy Liang, Michael I Jordan, and Dan Klein. 2009. Learning semantic correspondences with less supervision. In *Proceedings of ACL-47*, pages 91–99.

Ehud Reiter, Somayajulu Sripada, Jim Hunter, Jin Yu, and Ian Davy. 2005. Choosing words in computer-generated weather forecasts. *Artificial Intelligence*, 167(1):137–169.

Deb K Roy. 2002. Learning visually grounded words and syntax for a scene description task. *Computer Speech & Language*, 16(3):353–385.

Somayajulu Sripada, Ehud Reiter, Jim Hunter, and Jin Yu. 2002. Sumtime-meteo: Parallel corpus of naturally occurring forecast texts and weather data. *Computing Science Department, University of Aberdeen, Aberdeen, Scotland, Tech. Rep. AUCS/TR0201*.

Somayajulu Sripada, Ehud Reiter, and Ian Davy. 2003. Sumtime-mousam: Configurable marine weather forecast generator. *Expert Update*, 6(3):4–10.

Kees Van Deemter. 2012. *Not exactly: In praise of vagueness*. Oxford University Press.