Generating Justifications of Machine Learning Predictions

Or Biran
Columbia University
orb@cs.columbia.edu

Kathleen McKeown
Columbia University
kathy@cs.columbia.edu

Machine learning systems are increasingly used by humans to assist them in decision making. The systems produce predictions or recommendations which are then considered by a human decision-maker, and it is important that the prediction can be *justified*: the user will want to understand why the system produced its recommendation before making a decision.

For the rule-based expert systems that were common in past decades, it is often enough to *explain* how the system reached its decision by tracing the exact steps. Knowing how the system works helps the user decide whether the decision is justified. This is called the "glass box" or "white box" model, in contrast to the "black box" model where explanation is not given.

Recently, machine learning techniques have all but replaced rule-based methods, often resulting in increased accuracy and an ability to handle more complex problems. In contrast to rule-based expert systems, justifying the predictions of machine learning models is not a straightforward task: it is no longer the case that *explaining* how a prediction was reached automatically *justifies* it to the user. Due to the complex, quantitative and unintuitive nature of many machine learning models, it is unreasonable to expect that users who are not machine learning experts, even if they are experts in the domain of the prediction, will understand how the model works, regardless of how transparently it is presented. In other words, the glass box model is no longer useful for most users.

A black box with no justification at all, however, is even worse. We propose what might be called a "self-explaining box" model, where Natural Language Generation (NLG) is used to produce simple, short, qualitative and intuitive justifications for machine learning predictions, relying on the domain knowledge embodied in the features. Since the source of the generated text is the quantitative state of the model and the feature values, this task fits into the data-to-text generation paradigm.

Previous work shows that small, evidence-based justifications (in ML, evidence exists in the form of features) are more satisfactory to users than other types of explanations (Herlocker et al., 2000; Symeonidis et al., 2009; Papadimitriou et al., 2012) and that replacing numbers with linguistic qualifiers also enhances satisfaction (Herlocker et al., 2000; Lacave and Díez, 2002).

Our justifications, therefore, focus on a small number of features - those that are most relevant to the prediction - and present information about them in a qualitative rather than quantitative way.

The problems that are unique to this task arise in the *content planning* stage of NLG, as well as in earlier stages concerned with transforming the raw data into messages. In order to allow us to focus on these problems, our overall architecture aims to simplify other stages. In the remainder of this abstract we briefly describe our message structure, the architecture of our NLG system, and our approaches to solving the main subtasks.

1 Message structure

We use a shallow semantic structure, the Semantic Typed Template (STT). An STT is defined using a small semantic network of typed entity "slots" and relations among them. For example, the *prediction* STT contains an entity slot A of type *model*, an entity slot B of type *prediction*, and an entity slot C of unrestricted type, as well as the relations *madePrediction* $(A \rightarrow B)$ and *predicts* $(A \rightarrow C)$. In addition, each STT contains a set of paraphrasal templates. The prediction STT, for example, contains the template "the prediction for [C], made by [A], is [B]", among others. A Semantic Unit (SU), in turn, is an instantiated message, which corresponds to an STT and a set of concrete entities that have the types and relations specified by the STT.

We have created a core set of STTs for the jus-

<table>
<tr><td>Signal Analysis</td><td>Data Interpretation</td><td colspan="2">Content Selection</td><td>Discourse Planning</td><td colspan="2">Microplanning</td><td>Realization</td></tr>
<tr><td colspan="3">Feature Selection and Characterization</td><td>Feature Grounding</td><td colspan="2">Discourse Planning</td><td colspan="2">Templates</td></tr>
</table>

Table 1: Architecture comparison: our components shown below the pipeline of Reiter (2007)

tification domain (prediction, feature-has-role-in-prediction, etc). For this core set, the templates were manually specified. In addition, we are conducting research into extracting domain-specific STTs, along with sets of templates, from text corpora, as explained in section 3.

Using this representation, we can treat the messages as semantic during the content planning stage and then use a simple template approach in the microplanning and realization stages.

2 Architecture

Two problems arise that are unique to justification generation. The first is the problem of selecting the relevant features for the prediction and representing them qualitatively. We call the task of solving this problem *feature selection and characterization*. In our architecture, it corresponds to the *signal analysis* and *data interpretation* tasks of data-to-text generation, as well as to part of the *content selection* task of NLG in general.

The second problem is also related to the content selection task. In a justification, we want to present two types of content: the "core" content described above, discussing the state of the model and feature values that led to a prediction, and secondary content discussing the real-world interpretations of and relations among the features. While in online recommender systems the features are often simple enough that no further explanation is needed (e.g., the main actor of a movie), general ML models often have features that require some explanation. Think of features in medical diagnosis ("Erythema Nodosum"), financial price prediction ("EV/EBITDA") or network analysis systems ("Clustering Coefficient"). We call the task of solving this problem *feature grounding*.

In addition, we must also solve the other usual subtasks of NLG. While solutions to our two unique tasks together form our approach to content selection, we still must address *discourse planning*, *microplanning* and *realization*. Our approach to discourse planning, which we do not describe here for lack of space, also features simple *aggregation* (thereby contributing also to microplanning). The rest of the microplanning tasks, as well as realization, are left to the templates.

The overall architecture can be seen in Table 1, in comparison with the standard data-to-text architecture of Reiter (2007).

3 Component Research

In Biran and McKeown (2014), we describe in detail our approach to feature selection and characterization. Relying on two intuitive measures of a feature, its *effect* on the prediction and its *importance* in the model, we define a discrete role for each feature in the prediction. Based on these roles, we select the key features of the prediction (using one of multiple selection strategies) and produce simple messages that constitute the core of the content to generate.

For feature grounding, we have extracted a large taxonomic lexicon from Wikipedia (Biran and McKeown, 2013). The taxonomy allows us to produce messages describing the types of features, automatically discover groups of semantically related features (e.g., *valuation multiples* and *technical analysis signals* are major groups of common financial prediction features), and provides us with a lexicon of term synonyms.

In addition, we are planning to extract domain-specific definitional templates (as STTs) from Wikipedia in order to provide external definitions for features. For example, in the medical domain, features may be symptoms and medical conditions. Wikipedia contains information such as "*hypertension* can lead to *coronary heart disease*" and "*high blood pressure* is a major risk factor for *stroke*". Based on these and others, we can extract the *conditionAssociatedWithDisease* STT, with templates such as "[condition] is a major risk factor for [disease]", and the knowledge that this relation holds between the feature *hypertension* and the two diseases *coronary heart disease* and *stroke*. These extracted templates and related knowledge base can then be used to expand on a feature that has a role in the core set of messages. In that sense, our final planned NLG system can be said to be a hybrid of data-to-text and text-to-text approaches.

References

Or Biran and Kathleen McKeown. 2013. Classifying taxonomic relations between pairs of wikipedia articles. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing (IJCNLP)*, pages 788–794, Nagoya, Japan, October. Asian Federation of Natural Language Processing.

Or Biran and Kathleen McKeown. 2014. Justification narratives for individual classifications. In *Proceedings of the AutoML workshop at the International Conference of Machine Learning (ICML)*, Beijing, China, June.

Jonathan L. Herlocker, Joseph A. Konstan, and John Riedl. 2000. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work*, CSCW '00.

C. Lacave and F. J. Díez. 2002. A review of explanation methods for Bayesian networks. *Knowledge Engineering Review*, 17:107–127.

Alexis Papadimitriou, Panagiotis Symeonidis, and Yannis Manolopoulos. 2012. A generalized taxonomy of explanations styles for traditional and social recommender systems. *Data Min. Knowl. Discov.*, 24(3):555–583, May.

Ehud Reiter. 2007. An architecture for data-to-text systems. In *Proceedings of the Eleventh European Workshop on Natural Language Generation*, ENLG '07, pages 97–104, Stroudsburg, PA, USA. Association for Computational Linguistics.

Panagiotis Symeonidis, Alexandros Nanopoulos, and Yannis Manolopoulos. 2009. Moviexplain: A recommender system with explanations. In *Proceedings of the Third ACM Conference on Recommender Systems*, RecSys '09.