Information Sciences 255 (2014) 1–15

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A reconstruction decoder for computing with words

Dongrui Wu *

Machine Learning Lab, GE Global Research, Niskayuna, NY, USA

A R T I C L E I N F O

Article history:
Received 19 November 2012
Received in revised form 5 August 2013
Accepted 16 August 2013
Available online 2 September 2013

Keywords:
Computing with word
Perceptual Computer
Decoder
2-Tuple representation
Type-1 fuzzy sets
Interval type-2 fuzzy sets

A B S T R A C T

The Word decoder is a very important approach for decoding in the Perceptual Computer. It maps the computing with words (CWWs) engine output, which is a fuzzy set, into a word in a codebook so that it can be understood. However, the Word decoder suffers from significant information loss, i.e., the fuzzy set model of the mapped word may be quite different from the fuzzy set output by the CWW engine, especially when the codebook is small. In this paper we propose a Reconstruction decoder, which represents the CWW engine output as a combination of two successive codebook words with minimum information loss by solving a constrained optimization problem. The Reconstruction decoder preserves the shape information of the CWW engine output in a simple form without sacrificing much accuracy. It can be viewed as a generalized Word decoder and is also implicitly a Rank decoder. Moreover, it is equivalent to the 2-tuple representation under certain conditions. The effectiveness of the Reconstruction decoder is verified by three experiments.

1. Introduction

Computing with words (CWW) [47,48] is "*a methodology in which the objects of computation are words and propositions drawn from a natural language*". Usually the words and propositions are modeled by fuzzy sets (FSs) [46]. Many different approaches for CWW using FSs have been proposed so far [8,17,25,22,30,23,18,32,27,44,49,45,10,12,1,28,42,24]. According to Wang and Hao [31], these techniques may be classified into three categories:

(i) *The Extension Principle based models* [1,22,20,3], which operate on the underlying FS models of the linguistic terms using the Extension Principle [46]. Bonissone and Decker proposed the first such model in 1986 [1]. One of the latest developments is the Perceptual Computer (Per-C) [20,22], depicted in Fig. 1. It consists of three components: encoder, CWW engine and decoder. Perceptions (words) activate the Per-C and are the Per-C output (along with data); so, it is possible for a human to interact with the Per-C using just a vocabulary. The encoder transforms words into FSs and leads to a *codebook* – words with their associated FS models. Both type-1 (T1) and interval type-2 (IT2) FSs [19] may be used for word modeling. The outputs of the encoder activate a CWW engine, where the FSs are aggregated by novel weighted averages [39] or perceptual reasoning [38] according to the specific application. The output of the CWW engine is one or more other FSs, which are then mapped by the decoder into a recommendation (subjective judgment) with supporting data. Thus far, there are three kinds of decoders according to three forms of recommendations:
   (a) *Word:* To map a FS into a word, it must be possible to compare the *similarity* between two FSs. The Jaccard similarity measure [37] can be used to compute the similarities between the CWW engine output and all words in the codebook. Then, the word with the maximum similarity is chosen as the decoder's output.

* Tel.: +1 213 595 3269.
E-mail address: wud@ge.com

0020-0255/$ - see front matter
http://dx.doi.org/10.1016/j.ins.2013.08.050

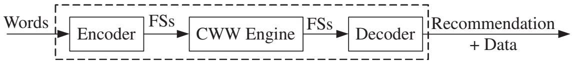

Fig. 1. Conceptual structure of the Perceptual Computer.

(b) *Rank:* Ranking is needed when several alternatives are compared to find the best. Because the performance of each alternative is represented by a FS obtained from the CWW engine, a ranking method for FSs is needed. A centroid-based ranking method for T1 and IT2 FSs is described in [37].

(c) *Class:* A classifier is necessary when the output of the CWW engine needs to be mapped into a decision category [21]. Subsethood [33,22,29] is useful for this purpose. One first computes the subsethood of the CWW engine output for each of the possible classes. Then, the final decision class is the one corresponding to the maximum subsethood.

(ii) *The symbolic model* [4,43,6], which makes computations on the indices of the linguistic terms. It first constructs an ordered linguistic term set, $\mathbb{W} = \{W_1, W_2, \ldots, W_N\}$, where $W_i < W_j$ if and only if $i < j$. Convex combination [4] is then used to recursively aggregate the terms. For example, to aggregate W_i and W_j with weight α and β, respectively, it computes

$$W_k = \frac{\alpha W_i + \beta W_j}{\alpha + \beta}, \tag{1}$$

where the term index k is determined as

$$k = i + round\left[\frac{\beta}{\alpha + \beta}(j - i)\right]. \tag{2}$$

To aggregate W_i, W_j and W_p with weight α,β and γ, respectively, i.e., to compute

$$W_{k'} = \frac{\alpha W_i + \beta W_j + \gamma W_p}{\alpha + \beta + \gamma} \tag{3}$$

it rewrites $W_{k'}$ as

$$W_{k'} = \frac{\alpha + \beta}{\alpha + \beta + \gamma} \cdot \frac{\alpha W_i + \beta W_j}{\alpha + \beta} + \frac{\gamma}{\alpha + \beta + \gamma} W_p = \frac{\alpha + \beta}{\alpha + \beta + \gamma} W_k + \frac{\gamma}{\alpha + \beta + \gamma} W_p \tag{4}$$

where W_k is the same as the one in (1) and k is computed by (2). $W_{k'}$ then becomes a two-term aggregation and k' is computed as

$$k' = k + round\left[\frac{\gamma}{\alpha + \beta + \gamma}(p - k)\right] \tag{5}$$

Aggregations involving more terms are computed in a similar recursive way. The intermediate results are numeric values, which must be approximated in each recursion to an integer in $[1,N]$ (e.g., k and k' above), which is the index of the associated linguistic term.

(iii) *The 2-tuple representation based model* [8,9,16,5,6], which is an improvement over the symbolic model. It was first proposed by Herrera and Martinez in 2000 [8] and followed by many others. Instead of representing the aggregation result as a single integer term index in $[1,N]$, it represents the result as a 2-tuple (W_n,α), where n is an integer linguistic term index, and $\alpha \in [-0.5, 0.5)$ is a numeric value representing the symbolic translation, i.e., the translation from the original result to the closest index label in the linguistic term set. More specifically, let $\mathbb{W} = \{W_1, W_2, \ldots, W_N\}$ be a linguistic term set and $\beta \in [1,N]$ be a value representing the result of a symbolic aggregation operation, then the 2-tuple representation (W_n,α) is computed as

$$n = round(\beta) \tag{6}$$

$$\alpha = \beta - n, \alpha \in [-0.5, .5) \tag{7}$$

As a result, the 2-tuple model allows a continuous representation of the linguistic information in its domain. Several aggregation operations using the 2-tuple representation, e.g., arithmetic mean, weighted average, ordered weighted average, have been developed [8].

Each category of models has its unique advantages and limitations. The Extension Principle based models can deal with any underlying FS models for the words, but they are computationally intensive. Moreover, their results usually do not match any of the initial linguistic terms, and hence an approximation process must be used to map the results back to the initial expression domain. This results in loss of information and hence the lack of precision [31,2]. The symbolic models