Toward building recommender systems for the circular economy

Exploring the perils of the European Waste Catalogue

van Capelleveen, G.; Amrit, C.; Zijm, H.; Yazan, D.M.; Abdi, A.

DOI
10.1016/j.jenvman.2020.111430

Publication date
2021

Document Version
Final published version

Published in
Journal of Environmental Management

License
CC BY-NC-ND

Link to publication

Citation for published version (APA):
van Capelleveen, G., Amrit, C., Zijm, H., Yazan, D. M., & Abdi, A. (2021). Toward building recommender systems for the circular economy: Exploring the perils of the European Waste Catalogue. *Journal of Environmental Management*, *277*, [111430]. https://doi.org/10.1016/j.jenvman.2020.111430

Journal of Environmental Management 277 (2021) 111430

Contents lists available at ScienceDirect

Journal of Environmental Management

journal homepage: http://www.elsevier.com/locate/jenvman

Research article

Toward building recommender systems for the circular economy: Exploring the perils of the European Waste Catalogue

Guido van Capelleveen [a,*], Chintan Amrit [b], Henk Zijm [a], Devrim Murat Yazan [a], Asad Abdi [a]

[a] *Department of Industrial Engineering and Business Information Systems, University of Twente, the Netherlands*
[b] *Faculty of Economics and Business Section Operations Management, University of Amsterdam, the Netherlands*

ARTICLE INFO

Keywords:
Recommender system
Tag recommendation
European waste catalogue (EWC)
Industrial symbiosis
Circular economy

ABSTRACT

The growth in the number of industries aiming at more sustainable business processes is driving the use of the European Waste Catalogue (EWC). For example, the identification of industrial symbiosis opportunities, in which a user-generated item description has to be annotated with exactly one EWC tag from an a priori defined tag ontology. This study aims to help researchers understand the perils of the EWC when building a recommender system based on natural language processing techniques. We experiment with semantic enhancement (an EWC thesaurus) and the linguistic contexts of words (learned by Word2vec) for detecting term vector similarity in addition to direct term matching algorithms, which often fail to detect an identical term in the short text generated by users. Our in-depth analysis provides an insight into why the different recommenders were unable to generate a correct annotation and motivates a discussion on the current design of the EWC system.

1. Introduction

One of the critical pathways to accelerate sustainable development is the reduction of waste emissions and primary resource use in resource-intensive industries. A mechanism contributing to this eco-innovation is industrial symbiosis, which is a cooperation between industries where the secondary outputs of one industry are utilized as (part of) primary inputs for the production processes of another industry (Chertow, 2000). The ontology used for annotating waste items in the European Union is the European Waste Catalogue (EWC) ("Commission Decision on the European List of Waste," 2000). This EWC ontology supports, for example, the identification of new symbiotic relations in eco-park development (Genc et al., 2019) and is used in information systems to relate waste stream characterization to implications on shipping, processing and disposal of waste (Pires et al., 2011). Specifically, these class labels, or EWC tags, typically assist users in searching for items of interest and help match users with corresponding interests. There are two key constraints for tagging the use of EWC codes (Gatzioura et al., 2019; van Capelleveen et al., 2018): (1) the tag needs to be selected from a defined a priori tag ontology consisting of 841 EWC codes, and (2) each item needs to be annotated with exactly one EWC code.

The tag recommender can be framed as a classical text-classification problem and therefore treated with supervised or unsupervised machine learning techniques (Khan et al., 2010), but the insufficient user-generated descriptions annotated with an EWC label at an initial stage makes it impossible to train any algorithm meaningfully (Gibert et al., 2018). To alleviate this common cold start problem, a content-based filtering approach that exploits the item description and EWC description is more suitable. The "noise" is the primary concern, omnipresent in many other tag systems, which needs to be dealt with in natural language (for example, ambiguity because of misspellings, synonymy, multilingualism, polysomy, hyponymy, hypernymy, idioms, etc.) (Golder and Huberman, 2006). To the best of our knowledge, issues related to the semantic matching of waste concepts are still poorly understood. Therefore, we explore the design of a method for this context (a quite short descriptive, highly jargon-based text) that can understand the syntactic, semantic, and contextual text similarity between user-generated item attributes and the EWC description. In our analysis, we test different vector initialization methods that include context and semantics in short text similarity measures and compare them to a baseline model that is based on a syntactic matching process. Our linguistic context of words is achieved through various configurations of Word2vec model learning (Google, 2019) and the semantic alternative suggestion is derived from an EWC thesaurus (U.K. Environment Agency, 2006). Furthermore, we perform an in-depth analysis of recommendations to find the root cause of success or failure of each

[*] Corresponding author.
E-mail address: g.c.vancapelleveen@utwente.nl (G. van Capelleveen).

https://doi.org/10.1016/j.jenvman.2020.111430
Received 19 March 2020; Received in revised form 4 September 2020; Accepted 25 September 2020
Available online 16 October 2020

recommender.

The remainder of the paper is organized as follows: Section 2 provides a brief overview of tag recommendation and the previous methods used to identify the context and semantics in both classical text classification and short text similarity measurements. In Section 3, we explain the methods that we tested in our experiment. Section 4 provides the results of the experiment, which includes a comparison of the method performance. Section 5 reflects on the effectiveness of the different recommender models and provides an interpretation of potential causes of recommender failure in addition to a discussion on the design of the EWC ontology for tag recommendation. Finally, Section 6 summarizes our findings and presents open issues for future work.

2. Background

2.1. Tag recommenders

There exists a variety of tag recommenders, each tailored to a specific domain of application, enforcing different restrictions, or including extensions, which make a tag recommender unique. Many of the design aspects that we are aware of in general recommender systems (van Capelleveen et al., 2019) can also be implemented in models for tag recommenders (e.g., personalization of tags (Hsu, 2013), context awareness of tags (Gao et al., 2019), and video content-based tags (Toderici et al., 2010)). A key aspect that makes the recommender problem unique is its composition and governance because the tag ontology structure and “language” is shaped by the way a tag set is governed. Because of the use of a fixed taxonomy in our tag recommender, its problem space shows close similarity with text classification. Methods, such as string similarity class prediction (Albitar et al., 2014), (large) scale text classification (Joulin et al., 2016; Partalas et al., 2015), or the most related, short text classification (Sriram et al., 2010) are the basis for building an EWC tag recommender algorithm.

2.2. Enhancing data for tag recommendation

Recently, tag recommendation has become an important subject of research as tags, among other textual features, have proven to be very effective in information retrieval tasks (Belém et al., 2017; Said and Bellogin, 2014). Several scholars have studied the methods of processing natural language and modeling the data that serves as an input for filtering algorithms with the main goal of improving the relevance of tag recommendation. Some studies indicate that incorporating item attributes from different categories improves the quality of tag suggestions (Hölbling et al., 2010; Zhao et al., 2010). In general, data such as abstract, keywords, and title (Ribeiro et al., 2015; Alepidou et al., 2011), and high-level concepts derived from various modalities such as geographical, visual, and textual information, complement each other. In combination, these attributes provide a richer context that can be used to improve tag relevance (Shah and Zimmermann, 2017). Another popular principle for alleviating tag noise is topic modeling. A topic model in a tagging context is a latent representation of a concept determined by the semantic relations between tags clustered around that concept (Zhong et al., 2017; Akther et al., 2012). External lexical databases that contain semantic relations (i.e., WordNet, Wikipedia) support the construction of these latent topic models. These databases allow us to create a contextual mapping between tags from a folksonomy, the existing taxonomy, and the latent concepts through the translation of syntactic representation by semantic relations (Qassimi et al., 2016; Zhang and Zeng, 2012; Wetzker et al., 2010; Subramaniyaswamy et al., 2013). The case in (Godoy et al., 2014) shows that semantic enhancement can also be used independently to increase the level of retrieval (recall/hit rate), supported by other works exploiting semantic databases, for example, WordNet and Wikipedia (Cantador et al., 2011; Subramaniyaswamy and ChenthurPandian, 2012), DBpedia (Ben-Lhachemi and Nfaoui, 2017; Mirizzi et al., 2010), classical search engine results and social tagging systems (Mirizzi et al., 2010).

2.3. Measuring short-text semantic similarity

The detection of similarity between short-text descriptions is found in various text-related similarity problems (e.g., conversational agents (Chakrabarti and Luger, 2015), linking questions with answers in Q&A systems (Wang and Varadharajan, 2007), plagiarism detection (Abdi et al., 2015), and text categorization (Ko et al., 2004)). These techniques are known as short-text semantic similarity (STSS) techniques and can be adapted to tag recommendation. STSS can be defined as a metric measuring the degree of similarity between pairs of small textual units, typically with a length of less than 20 words, ignoring the grammatical correctness of a sentence (O'Shea et al., 2008). Short contexts rarely have common words in the exact lexical composition: hence, the key challenge of STSS is to overcome the detection of the right semantics and context. Methods that can detect these short-text similarities are string-based, corpus-based, and knowledge-based similarity measures (Gomaa and Fahmy, 2013).

Although there is no unified method that applies best to every context, an appropriate method can be selected by evaluating the data in the context of the application. Testing the effect of including each aspect of a contextual or semantic relation leads to a better algorithm design for that context. There is extensive literature on what type of aspects and associated methods can be dealt with, (Khan et al., 2010; Pedersen, 2008), including but not limited to boundary detection (e.g., sentence splitting, morphological segmentation, tokenization, topic segmentation), grammar induction, lexical semantics (synonymy, multilingualism, polysomy, hyponymy, hypernymy, and idioms), text representation (nouns, adjectives), word sense disambiguation (contextual meaning), text sanitation (e.g., removing stopwords, spelling corrections, and noisy data), deriving a word to a common base form (stemming and lemmatization), recognition tasks (e.g., terminology extraction, named entity recognition), collocations, sentiment analysis, text enrichment (e.g., including a title or keywords), expanding to second order similarity (e.g., word expansion, context augmentation, fuzzy matching short-text replacements), and distributional semantics (e.g., word embeddings).

3. Methodology

The design of a novel approach to identify waste concepts in short-text waste descriptions to generate and evaluate tag recommendations follow the design science research approach (Hevner et al., 2004) and are guided by the principles of (Peffers et al., 2007). First, the problem definition is provided, followed by an explanation of the data characteristics. Then, we present the experimental setup and introduce our model.

3.1. Problem definition

The problem of EWC tag recommendation is formally defined as follows. There is a tuple $F := (U, I, T, A)$, which describes the users U, the items I, the tags T, and the assignment of a tag by a ternary relation between them, that is, $A \subseteq U \times I \times T$. For a user $u \in U$ and a given item $i \in I$, the problem, to be solved by a tag recommender system, is to find a tag $t(u, i) \in T$ for the user to annotate the item (Godoy and Corbellini, 2016; Belém et al., 2017). Each $i \in I$ contains a short text string i_{desc} (typically less than 20 words) describing a waste. This set of text strings for all items is denoted by $I_{desc} = \{i_{desc} | i \in I\}$. The tag vocabulary T consists of tags where each $t \in T$ represents a unique EWC code (at the third level in the EWC ontology; see also Table 2) from the European Waste Catalogue (“Commission Decision on the European List of Waste,” 2000). Each tag t has two components: the EWC code t_{ewc} and the EWC description t_{desc}. Another useful concept is the bag of words (BOW), which is a sequence of keywords extracted from a general text string.

Such a keyword representation b_i for an item description using the BOW concept is derived from i_{desc}. Similarly, a keyword-based representation for a tag description b_t using the BOW is derived from t_{desc}. Finally, a keyword-based representation of term synonyms for tag description $b_{syn,t}$ is derived from a thesaurus (described in Section 3.4). The collection of associated BOW pairs (each pair combining the keyword-based representations of an item and a tag) is denoted as (b_i,b_t). A restriction to our tag annotation problem is that there must be one and only one b_t associated with a b_i (i.e., an item is annotated by one and only one EWC tag). In contrast to other tag recommendation problems in which the association can be expressed on a numerical scale (such as with ratings), the association between b_t and b_i is expressed as a binary value, which means that the assigned EWC label is either correct or incorrect.

3.2. Data characteristics of the "waste" domain

There are six data sources used in the experiment. The first three data sources are characteristics of the waste domain. These are (a) data for prediction and evaluation, (b) the EWC tag ontology, and (c) the EWC tag thesaurus. The other three sources are training corpora for Word2vec (see Section 4.1).

First, we use data that consist of (1) waste descriptions specifying a waste item for which we intend to recommend the EWC tag and (2) the correct class label (i.e., the EWC tag) that can be used to evaluate the recommender algorithm. These data originate from industrial symbiotic workshops that were part of the EU-funded SHAREBOX project (Sharebox Project, 2017), providing a variety of waste items with associated resource interests from industry. Waste items are often described using short sentences, or only a few keywords, commonly with less than 10 words. These items have been annotated with an EWC tag. The EWC tag (i.e., the description at level 3 in the EWC ontology) can be seen as the class label that serves to test the prediction task of the recommender system. The evaluation data set is imbalanced, which means the classes are not represented equally in the test data. The example data are shown in Table 1.

Next, we use the EWC ontology. This ontology is a statistical classification system defined by the European Commission to be used in reporting waste statistics in the European Union in a concise manner ("Commission Decision on the European List of Waste," 2000). Each EWC code represents a class that refers to a group of waste materials. The EWC code is the label of this class and is used to annotate items in the evaluation data. An example of such an EWC structure is shown in Table 2.

Finally, we use EWC thesaurus data in two of the suggested methods (see Section 3.3) to enrich the descriptions from the EWC ontology. The selected thesaurus is derived from the guidance document on the "List of Wastes" composed by the UK Environmental Agency (see (U.K. Environment Agency, 2006), Annex 1). An example of the thesaurus data is shown in Table 3.

Table 1
Example of evaluation data (Sharebox) (Sharebox Project, 2017). The full test set contains 311 entries describing materials, each item description has a mean of 4.945 terms per entry with a standard deviation of 3.040. There are 691 unique terms in the item descriptions before pre-processing, and 479 unique terms after pre-processing. The classes assigned to the entries are imbalanced over the data set.

Item description	EWC annotation
Iron and steel slag: Concrete tiles can be taken as one of the main components	15 01 04
Sawmill dust and shavings	03 01 04
Food and textile waste	02 02 03

Table 2
Sample illustrating the structure of the waste classification system EWC (ontology) ("Commission Decision on the European List of Waste," 2000). The full data set contains 841 classes (at level 3), each EWC description has a mean of 4.753 terms with a standard deviation of 2.038. There are 628 unique terms in the EWC descriptions before pre-processing, and 475 unique terms after pre-processing.

Chapter (Level 1)	Sub Chapter (Level 2)	Full Code (Level 3)	Description
03			Wastes from wood processing and the production of panels and furniture, pulp, paper and cardboard
	03 01		Wastes from wood processing and the production of panels and furniture
		03 01 01	Waste bark and cork

Table 3
Example of thesaurus data (U.K. Environment Agency, 2006). The full data set contains 691 classes (at level 3), which means there are several EWC codes that do not have a thesaurus description, each EWC thesaurus description has a mean of 8.794 terms with a standard deviation of 6.452. There are 1503 unique terms in the EWC thesaurus description before pre-processing, and 1232 unique terms after pre-processing.

EWC	Thesaurus entry
15 01 04	Cans - aluminium, Cans - metal, Metal containers - used, Aluminium, Aluminium cans, Aerosol containers - empty, Drums - steel, Steel drums, Aluminium foil, Containers (metal) - used, Containers - aerosol - empty, Containers - metal (contaminated), Contain
03 01 04	Chipboard, Sawdust, Sawdust - contaminated, Shavings - wood, Timber - treated, Dust - sander, Hardboard, Wood, Wood cuttings
02 02 03	Food - condemned, Condemned food, Food processing waste, Animal fat, Fish - processing waste, Fish carcasses, Kitchen waste, Meat - unfit for consumption, Poultry waste, Shellfish processing waste, Pigs, Cows, Sheep

3.3. Experiment setup

To test the algorithm based on an already "tagged" item set, we attempt to assign a tag to each item using different methods and validate the correctness of this assignment. The research setup is illustrated in Fig. 1. A prerequisite for all text, the short texts from each of the evaluation data, EWC ontology, and thesaurus data is that the natural language is pre-processed and converted into a BOW. This procedure consists of a sequence of steps, as follows:

1. All the characters are converted to lowercase and all numbers and special characters are removed.
2. All items are tokenized into the BOW.
3. All terms that contain less than 4 characters are removed (this removes most non-words without much meaning, as well as abbreviations)
4. All stop words "English" from the Natural Language Tool Kit (NLTK) are removed (Natural Language Tool Kit, 2019). A stop word refers to the most frequently used word (such as "the", "a", "about", and "also") that are filtered before or after the NLP.
5. All words are stemmed using the empirically justified Porter algorithm (Porter, 1980) in NTLK (Natural Language Tool Kit, 2019). Stemming is the process of removing the inflectional forms and sometimes the derivations of the word, by identifying the morphological root of a word (Manning et al., 2008).
6. All common nonsignificant stemmed terminology used in industrial symbiosis are removed. These include *"wast", "process", "consult", "advic", "train", "servic", "manag", "management", "recycl", "industri", "materi", "quantiti", "support", "residu", "organ", "remaind".*

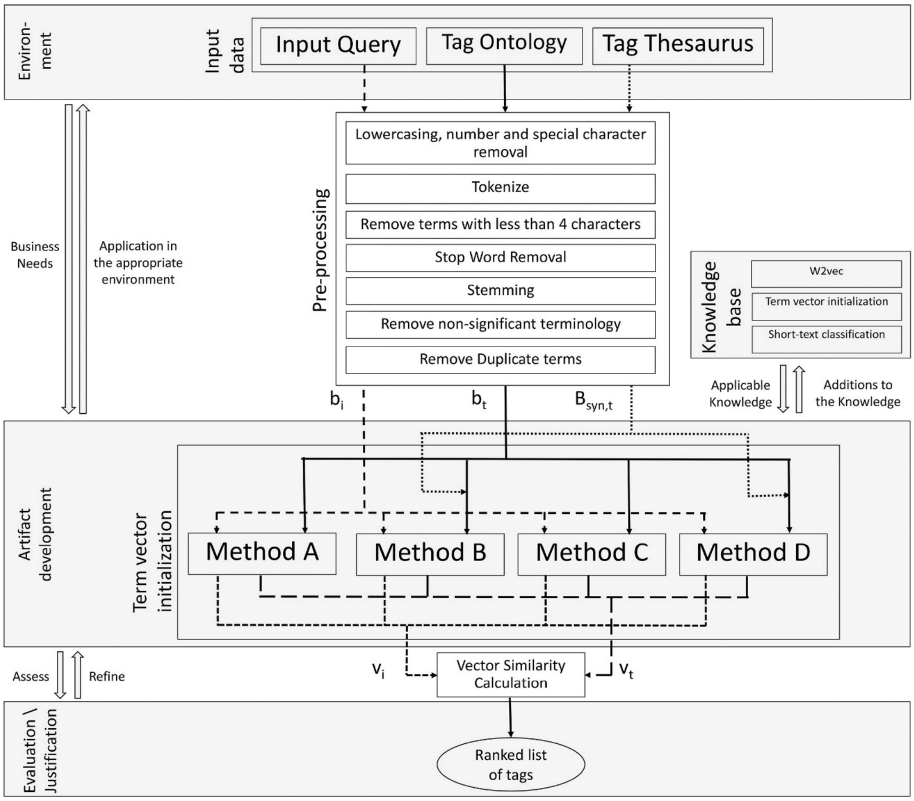
Environ-
ment
Input
data
Input Query
Tag Ontology
Tag Thesaurus
Pre-processing
Lowercasing, number and special character removal
Tokenize
Remove terms with less than 4 characters
Stop Word Removal
Stemming
Remove non-significant terminology
Remove Duplicate terms
Business Needs
Application in the appropriate environment
Knowledge base
W2vec
Term vector initialization
Short-text classification
Applicable Knowledge
Additions to the Knowledge
b$_i$
b$_t$
B$_{syn,t}$
Artifact development
Term vector initialization
Method A
Method B
Method C
Method D
v$_i$
Vector Similarity Calculation
v$_t$
Assess
Refine
Evaluation \ Justification
Ranked list of tags

Fig. 1. Design methodology.

7. All duplicate terms were removed from the BOW.

After this set of pre-processing steps, the remainders of the BOW form the terms that are adopted in the term vector. This works as follows. We first replace the now pre-processed BOW's b_i and b_t by numerically valued vectors v_i and v_t with length equal to the total number of unique terms appearing in either b_i or b_t, and set its elements $v_{i,j}(v_{t,j})$ equal to one if the j-th term appears in $b_i(b_t)$, and zero otherwise (this is called the term vector initialization). Note that a basic short-text similarity technique that relies on term vector initialization (Method A), as in the example above, cannot detect similarity if there are no shared terms. To mitigate this problem, we propose different term vector initialization methods (Methods B, C, and D), which use pseudo-semantic similarity of the terms (described in Section 3.4).

Before calculating the similarity score between the vectors v_i and v_t we normalize the vectors v_i and v_t, using feature scaling (see Zheng and Casari (2018)), that is, scaling back the magnitudes of the vectors to the range $[0,1]$ without changing the direction of the vector. Then, a similarity score is calculated to predict the classification tag b_t. We selected the classic cosine similarity measure (see Equation (1)) for measuring the term similarity in all four methods, as it is a robust similarity measure (Huang, 2008).

$$s_{i,t} = CosSim(v_i, v_t) = \frac{\sum_j v_{i,j} v_{t,j}}{\sqrt{\sum_j v_{i,j}^2}\sqrt{\sum_j v_{t,j}^2}} \tag{1}$$

Equation (1) defines the similarity score $s_{i,t}$ which is calculated by the cosine similarity measure *CosSim* between two vectors v_i and v_t of equal length, where $v_{i,j}$ and $v_{t,j}$ denotes the elements of the term vectors v_i and v_t, respectively. A tag prediction task on item i results in a ranked list consisting of a similarity score for each tag for that item i, denoted as $R_i = \{r_{i,1}, r_{i,2}, \ldots, r_{i,|T|}\}$, where $|T|$ is the cardinality of T, and $r_{i,t} = (t, s_{i,t})$ with t a specific tag and $s_{i,t}$ the similarity score of that tag on item i. The list is cut off for evaluation by taking the top k results from R_i by ranking s_i in a descending order.

3.4. Proposed methods

3.4.1. Method A (base): basic term matching

In this method we simply create the binary-valued vectors v_i and v_t from the corresponding pre-processed BOW b_i and b_t as discussed above (see example Table 4).

3.4.2. Method B (thesaurus): basic term matching, enriched with a thesaurus (adding semantics)

The second method applies in-direct term matching with the support of a semantic enhancement technique, that is, a thesaurus. The EWC thesaurus E is exploited to increase the number of semantic links between the waste item description i_{desc} and the correct tag description t_{desc}. These semantically equivalent terms for the purposes of information retrieval are called synsets. The EWC thesaurus consists of a reference dataset $E = \{(t, e_{syn,t})\}$ where t is an EWC tag, and $e_{syn,t}$ is the associated synset. These synsets are derived from the 'List of Wastes' (U. K. Environment Agency, 2006). For each tag t, the associated synset $e_{syn,t}$ is also pre-processed using the steps previously described, to compose BOW $b_{syn,t}$. For each tag t, we expand the BOW b_t with the BOW $b_{syn,t}$.

Table 4
Example term vectors (Method A).

	b_i		b_t	
	food	textil	unsuit	consumpt
v_i	1	1	0	0
v_t	0	0	1	1

Item entry i_{desc}: *"Food and textile waste"*. Tag entry t_{desc}: *"materials unsuitable for consumption or processing" (02 02 03)*. Note: the terms *"for, or, and" (stop words), and "materials, waste, processing" (non-significant terminology)* are removed in pre-processing.

Then, similar to the basic term matching technique (Method A), each BOW b_i is converted to a vector v_i and, similarly, each BOW b_t is converted to a vector v_t (see Table 5).

3.4.3. Method C (Word2vec): using Word2vec models (adding context)

The third method is based on Word2vec. Word2vec (W2V) (Google, 2019) is a two-layer neural net that processes text in order to produce word embeddings, which are vector representations of a particular word that captures the context of a word in a text. This technique can be used to calculate the similarity between words based on the context as an alternative or as an addition to syntactic or semantic matching techniques. In order to train how similar terms are in that context, Word2vec requires a large text corpus. Three configurations are tested, from which the best one is selected (see Section 4.1). The purpose of testing different configurations is to find a well-performing configuration that can lead to a better understanding of the use of the Word2vec approach in retrieving more or alternative, albeit correct tags for EWC tag recommendation. The three configurations are differentiated by the corpus used to train Word2vec, which essentially determines which associations can be retrieved. A heuristic set of hyper-parameter settings, each tailored to that corpus, is used (and explained in Section 4.1). The three configurations are as follows:

- **W2V Configuration "*Google News*":** The first configuration is based on the well-known pre-trained Google News Word2vec file (Google, 2019).
- **W2V Configuration "*Common Crawl*"::** The second configuration uses a pre-trained Word2vec file on one of the largest text corpora existing today, the Common Crawl (2019), which is offered by the fastText library of Facebook (Facebook Inc., 2019).
- **W2V Configuration "*Elsevier*":** The third configuration is based on a "waste" data corpus manually constructed by scraping the "abstract" and the "introduction" section from the available academic papers indexed by Elsevier Scopus (Elsevier, 2019) retrieved through the search query "waste". Then, the Word2vec model is trained using the Word2vec implementation in the Gensim library (Řehůřek, 2019).

After training the model, we can assign Word2vec similarities to the dimensions of the term vector. To do that, we first create a direct term vector using the vector creation procedure as described above, where each non-zero element is multiplied with a so-called *term_weight* value to determine the balance between the direct term matching and the context-based Word2vec similarity scores. The latter scores are calculated by Word2vec to indicate the degree of similarity between two terms that appear in b_i and b_t, respectively. Two approaches are tested to represent the context-based term similarity created by the Word2vec similarity score(s), which are named the *"Average W2V Similarity"* and the *"Maximum W2V Similarity"*. Both methods are illustrated with an example in Table 6.

- **Term vector configuration: '*Average W2V Similarity*':** The *"Average W2V Similarity"* is an adaptation of the similarity measure in (Croft et al., 2013). This approach replaces each zero in the direct term vector with the average of all the Word2vec similarity values between the term of that dimension and each of the initially non-zero terms of the other vector.
- **Term vector configuration: "*Maximum W2V Similarity*":** The *"Maximum W2V Similarity"* is an adaptation of the similarity measure in (Crockett et al., 2006). This approach replaces each zero in the direct term vector with the maximum Word2vec similarity value between the term of that dimension and each of the initially nonzero terms of the other vector.

Table 6
Examples term vectors (Method C).

	b_i		b_t		
	powder	coat	paint	varnish	mention
Direct Term Vector:					
Term vector v_i	2	2	0	0	0
Term vector v_t	0	0	2	2	2
W2V values:					
W2V weight "powder"	X	X	.355	.379	.076
W2V weight "coat"	X	X	.452	.491	.105
W2V weight "paint"	.355	.452	X	X	X
W2V weight "varnish"	.379	.491	X	X	X
W2V weight "mention"	.065	.058	X	X	X
Term Vector Config.:					
Average value	.266	.334	.404	.435	.091
Maximum value	.379	.491	.452	.491	.105
Weighted Term Vector:					
Weighted v_i using Average	2	2	.404	.435	.091
Weighted v_t using Average	.266	.334	2	2	2
Weighted v_i using Maximum	2	2	.452	.491	.105
Weighted v_t using Maximum	.379	.491	2	2	2

Item entry i_{desc}: "Powder coating waste". Tag entry t_{desc}: "Waste paint and varnish other than those mentioned in 08 01 11" (08 01 12). Term weight *term_weight*: 2. Note: the terms *"and, in, other, than, those" (stop word), "waste" (non-significant terminology), and "08 01 11" (numbers)* are removed during pre-processing.

In case Word2vec fails to provide a similarity score for some terms, we use the score of Ratcliff/Obershelp pattern recognition (Ratcliff and Metzener, 1988). Ratcliff/Obershelp computes the similarity between two terms based on the number of matching characters divided by the total number of characters in the two terms.

3.4.4. Method D (Word2vec thesaurus): using Word2vec models (adding context), enriched with a thesaurus (adding semantics)

Method D is a combination of Method B and Method C. It employs the thesaurus as used in the Method B, thereby, enriching the dataset with a larger variety of terms. After obtaining the term vectors v_i and v_t using method B, we apply Method C to include the Word2vec values in the term vector which together create the weighed vectors v_i and v_t.

Table 5
Example term vectors (Method B).

	b_i		b_t													
	food	textil	unsuit	consumpt	condemn	anim	fish	carcass	kitchen	meat	unfit	poultri	shellfish	pig	cow	sheep
v_i	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
v_t	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Item entry i_{desc}: *"Food and textile waste"*. Tag entry t_{desc}: *"materials unsuitable for consumption or processing" (02 02 03)*. Thesaurus entry e_{ewc}: *"Food - condemned, Condemned food, Food processing waste, Animal fat, Fish - processing waste, Fish carcasses, Kitchen waste, Meat - unfit for consumption, Poultry waste, Shellfish processing waste, Pigs, Cows, Sheep"* Note: the terms *"for, or, and" (stop words), "materials, waste, processing" (non-significant terminology), and "fat" (less than 4 characters before stemming)* are removed during pre-processing.

4. Algorithm performance test

To assess the performance of the proposed set of methods and to provide comparative measures, an experiment was conducted. First, the configuration settings for training the Word2vec models are explained. This is followed by a set of definitions of the evaluation metrics and the optimization process for the parameters of the tag recommender. Finally, the methods are compared given their optimal parameter configurations.

4.1. Word2vec model and hyperparameters

Three different configurations (see Table 7) were tested to find a strong Word2vec model to represent our proposed context-based approaches (Methods C and D).

The first configuration is based on the "Google News" data set which is a well-known baseline for Word2vec experiments, but not optimized for jargon. To align with the semantic use of rare terms and jargon, we created two other configurations: that is, the "Common Crawl" and the "Elsevier" configuration. The "Common Crawl" configuration is based on one of the largest data sets on which Word2vec has been globally trained, and it uses the latest continuous bag of word techniques implemented by Facebook. The pre-trained Word2vec model uses fastText which exploits enriched word vectors with subword information to optimize the performance of rare words. This is because fastText can compute word representations for words that did not appear in the training data (Bojanowski et al., 2017). The "Elsevier" configuration was created from scratch. The key is to construct a more jargon containing data set to train Word2vec. We collected data from Elsevier Scopus through a search of abstracts and introductions of academic papers that are related to the search query "waste". Of the more than $8e^{+5}$ indexed documents, we were able to retrieve $3.57e^{+5}$ documents, for which the raw text content was downloaded through the Scopus API. We removed the numbers, links, citations, and page formatting hyphenations so that only the clean sentences remain. We trained the Word2vec using this data and the configuration settings as noted in Table 7. Because the Elsevier dataset is targeted to maximize the retrieval of jargon used in the waste domain, we configured the parameters to support this accordingly. The number of dimensions of learning word embeddings is set to 1000 (where the rule of thumb is 50–300 for a good balance in performance), and the minimal term count is lowered to 1, which should favor learning word embeddings for rare terms (Patel and Bhattacharyya, 2017; Mikolov et al., 2013). We use the heuristic of 10 (pos/neg) for the window size (based on (Pennington et al., 2014)). The window size determines how many words before and after a given word are included in the context that Word2vec uses to learn relationships. A larger window size typically tends to capture more domain context terms whereas a smaller window size captures more dependency-based embeddings (e.g., the words that are most similar, synonyms or direct replacements of the originated term).

Table 7
Description of the data, and the Word2vec model hyperparameter settings.

	Google (Google, 2019)	Common Crawl (Grave et al., 2018)	Elsevier (Elsevier, 2019)
No. terms/pag.	100E9 words	1.5E11 pages	3.57E5 docs
File Size	NA	234 TiB	2.41 GB
Training Platform:			
Framework	Word2vec (Google)	Word2vec (fastText)	Word2vec (Gensim)
Model configuration:			
Architecture	Continuous Skip-gram	CBOW	Continuous Skip-gram
File size	3.6 GB	7.1 GB	4.9 GB
Word embedding (Dimension size)	300	300	1000
Epochs	NA	5	5
Minimal term count	5	NA	1
Window size	5pos/15 neg	5 pos/10 neg	10pos/10neg
N-gram	–	5	–

4.2. Evaluation metrics

A series of commonly applied evaluation metrics are used to evaluate and compare the performance of our proposed methods for tag recommendation (Said and Bellogín, 2018; Ekstrand et al., 2011). All the experiments were conducted in an off-line setting. Our EWC tag problem is a multi-class classification problem in which the EWC code acts as the class label. A multi-class classification problem is a classification task with more than two classes under the assumption that each item is assigned one and only one class label. Multi-class data, such as ours, are often imbalanced, which means that the classes are not represented equally. Therefore, the evaluation uses metrics at both the micro and macro levels (Scikit-learn developers, 2020). We measured the precision, recall, accuracy, F-measure (or F1), mean reciprocal rank (MRR), and discounted cumulative gain (DCG). We measure precision, recall, accuracy, and F1 at the micro-level because micro-averaging treats all classes by their relative size, which better reflects imbalanced data subsets. Furthermore, we measure the balanced accuracy at the macro level. The Greek letters μ, *M*, and γ indicate micro-, macro-, and balanced-averaging, respectively. The other metrics *MRR* and *DCG* are rank-based metrics. Rank-based metrics measure a ranked list using a function that discounts a correct tag when found at a lower position in the list.

First, we create a matrix P with a prediction $s_{i,t}$, obtained from the similarity measure $CosSim()$ in Equation (1), between each item i and each EWC tag t. From P, we can obtain the ranked list of predictions R_i (see Section 3.3). The precision, recall, accuracy, and F-measure are metrics measured at the list level. A metric at the list level measures a list at a particular k, meaning that if the correct tag is among the first k ranked tags, it is considered as a correct list of tags and thus a correct recommendation (Baeza-Yates and Ribeiro-Neto, 2011). The relevance of tag prediction is evaluated in a binary fashion; that is, a list of tag predictions can only be classified as correct or incorrect recommendation. This means that, for every i, the recommendation is a list of the k highest predictions $(s_{i,t_1}, s_{i,t_2}, \dots, s_{i,t_k})$ in R_i, where k is the maximum number of ranked tags. First, we define the class-level metrics, and then we show how the list-level metrics at the micro and macro levels are defined, and finally, we present the rank-based metrics.

4.2.1. Tag- or class-level metrics

The set of tags T (i.e., level 3 EWC descriptions) represents the classes for evaluating our multi-class tag assignment problem. In the evaluation of a binary classification, tp denotes the number of true positives, fp is the number of false positives, tn is the number of true negatives, and fn is the number of false negatives. In multi-class evaluation, we use a class-based definition of true positives tp, false positives fp, false negatives fn, and true negatives tn. We use t to indicate that the counts of tp,fp, fn, and tn are measured for a specific class (i.e., tag). A true positive tp_t for a tag t is when the correct tag label is class t and the list of predicted tag labels contains the correct tag label. A false positive fp_t for a tag t is when the correct tag label is not class t and a list of predicted tags contains the correct tag label. A false negative fn_t for a tag t is when the correct tag label is class t, and the list of predicted tag labels does not contain the correct tag label. A true negative tn_t for a tag t is when the correct tag label is not class t, and a list of predicted tags does not contain the correct tag label.

The precision related to tag t (Equation (2)) is the number of correctly recommended tags t divided by all instances in which tag t has been recommended.

$$Precision_t = \frac{tp_t}{tp_t + fp_t} \tag{2}$$

The recall related to tag t (Equation (3)) is the number of instances that are correctly tagged with t divided by the total number of instances for which tag t would be the correct label.

$$Recall_t = \frac{tp_t}{tp_t + fn_t} \tag{3}$$

The F-measure (or F_1 score) (Equation (4)) is a different measure of a test's accuracy that evaluates the precision and recall in a harmonic mean.

$$F - measure_t = 2 \cdot \frac{Precision_t \cdot Recall_t}{Precision_t + Recall_t} \tag{4}$$

Accuracy (Equation (5)) is the fraction of measurements of correctly identified tags as either truly positive or truly negative out of the total number of items.

$$Accuracy_t = \frac{tp_t + tn_t}{tp_t + tn_t + fp_t + fn_t} \tag{5}$$

4.2.2. Micro-level metrics

When we evaluate the classes at a micro level, we count all tp_t, tn_t, fp_t and tn_t globally, that is, summing the values in the numerator and denominator prior to division. Equation (6) denotes the micro precision as used for multi-class classification, that is, a generalization of the precision for multiple classes t.

$$Precision_\mu = \frac{\sum_{t \in T} tp_t}{\sum_{t \in T} (tp_t + fp_t)} \tag{6}$$

Equation (7) denotes the micro-recall, as used for multi-class classification, that is, a generalization of the recall for multiple classes t.

$$Recall_\mu = \frac{\sum_{t \in T} tp_t}{\sum_{t \in T} (tp_t + fn_t)} \tag{7}$$

Equation (8) denotes the micro F-measure (or micro F_1 score) as used for multi-class classification, that is, a generalization of the F-measure for multiple classes t.

$$F - measure_\mu = 2 \cdot \frac{Precision_\mu \cdot Recall_\mu}{Precision_\mu + Recall_\mu} \tag{8}$$

Equation (9) denotes the micro accuracy as used for multi-class classification, that is, a generalization of the accuracy, defined over all tags. The micro accuracy is the fraction of correct predictions over the total number of predictions (regardless of the positive or negative label).

$$Accuracy_\mu = \frac{\sum_{t \in T} (tp_t + tn_t)}{\sum_{t \in T} (tp_t + tn_t + fp_t + fn_t)} \tag{9}$$

As the tn_t inflates the accuracy when there are many classes T, we typically measure (see (Scikit-learn developers, 2020)) the test result from the perspective of the correct class t for a multi-class evaluation. Therefore, the modified micro accuracy is the fraction of correct predictions for a class t over the number of predictions for items for which the correct tag label is t, as denoted in Equation (10), which is similar to micro recall.

$$ModifiedAccuracy_\mu = \frac{\sum_{t \in T} tp_t}{\sum_{t \in T} (tp_t + fn_t)} = Recall_\mu \tag{10}$$

4.2.3. Macro-level metrics

When we evaluate all classes at a macro level, we calculate the metrics for each class t, and then calculate the (unweighted or weighted) mean. Equation (11) denotes the unweighted macro accuracy (Sokolova and Lapalme, 2009).

$$Accuracy_M = \frac{1}{|T|} \sum_{t \in T} \frac{tp_t + tn_t}{tp_t + fn_t + fp_t + tn_t} \tag{11}$$

In practice, as the tn_t inflates the accuracy when there are many classes T, we typically measure a balanced score instead. Equation (12) denotes balanced accuracy (Scikit-learn developers, 2020), as used for multi-class classification which avoids inflated performance estimates on imbalanced datasets. Balanced accuracy is the macro-average of the recall scores.

$$BalancedAccuracy_\gamma = \frac{1}{|T|} \sum_{t \in T} Recall_t = Recall_M \tag{12}$$

However, in the implementation of sklearn (Scikit-learn developers, 2020), the balanced accuracy (see Equation (13)) does not count the scores for classes that did not receive predictions. Let X denote the set of all tags that have been predicted at least once, with $|X|$ the cardinality of X. Then we have:

$$ModifiedBalancedAccuracy_\gamma = \frac{1}{|X|} \sum_{t \in X} Recall_t \tag{13}$$

4.2.4. Rank-based Metrics

Finally, we define rank-based evaluation metrics. The MRR (see Equation (14)) is a rank measure appropriate for evaluating a rank in which there is only one relevant result, as it only uses the single highest-ranked relevant item (Baeza-Yates and Ribeiro-Neto, 2011). This metric is similar to the average reciprocal hit-rank (ARHR), defined in (Deshpande and Karypis, 2004). Let R be the set of all recommended lists, and let Z denote the set of all items for which the recommended list of tags contains at least one correct tag in the top k tags. Define $\boldsymbol{rank_i}$ as the position of the correct tag in Z. Then, the MRR is defined as

$$MRR = \frac{1}{|R|} \sum_{i \in Z} \frac{1}{rank_i} \tag{14}$$

Another useful evaluation metric is the DCG (Järvelin and Kekäläinen, 2002). In recommender systems, relevance levels can be binary (indicating whether a tag is relevant or irrelevant for an item) or graded on a scale (indicating a tag has a varying degree of relevance with an item, e.g., 4 out of 5). In our evaluation, the relevance score on an item is a binary graded relevance score of a tag (i.e., a tag is either correct or incorrect). The DCG_i (Equation (15)) is a cumulation of all binary graded relevance scores from position 1 to k in the ranked list of tags for an item i, with $\boldsymbol{rel_{i,p}}$ the binary graded relevance score at position p, which is discounted by a log function in the denominator that progressively reduces the impact of a correctly ranked tag when found at a lower ranked position.

$$DCG_i = \sum_{p=1}^{k} \frac{2^{rel_{i,p}} - 1}{log_2(p+1)} \tag{15}$$

In our case each recommended list contains at most one correct tag, which reduces to the following (see Equation (16)):

$$DCG_i = \begin{cases} 0, & \text{if the recommended list contains no correct tag} \\ \frac{1}{log_2(p+1)}, & \text{if the correct tag is found at position } p \end{cases} \tag{16}$$

The DCG (Equation (17)) is the mean of the DCG_i of all recommended lists in R.

$$nDCG = \frac{1}{|R|} \sum_{i \in Z} DCG_i \tag{17}$$

While in many recommender evaluations it is common to provide a normalized discounted cumulative gain (nDCG), this does not apply to binary graded multi-class evaluation. This is because the DCG_i would be normalized over the same ideal ranking. Let the ideal discounted cumulative gain $IDCG_i$ (Equation (18)) represent the maximum possible DCG_i. This $IDCG_i$ is calculated as DCG_i for the sorted list with a maximum length of positions k of all relevant tags ordered by their binary graded relevance score, with k as the number of predicted tags and $rel_{i,p}$ as the binary graded relevance score of the tag at position p.

$$IDCG_i = \sum_{p=1}^{k} \frac{2^{rel_{i,p}} - 1}{log_2(p+1)} = 1 \tag{18}$$

The $nDCG_i$ (see Equation (19)) for a single ranked list of tags for item i is denoted as the fraction of DCG_i over the $IDCG_i$.

$$nDCG_i = \frac{DCG_i}{IDCG_i} = \frac{DCG_i}{1} = DCG_i \tag{19}$$

Because the gain is only achieved at the correct tag, which is the first position in a ranked list of tags, normalization is irrelevant as the *IDCG* is then equal to 1. Therefore, we adopted the *DCG* instead of *nDCG* in our experiment.

4.3. Parameter setting of the recommenders

A few selection criteria are applied as a prerequisite for our comparison in Section 4.4. First we need to find the optimal initialization for two parameters. Parameter (1) determines the *term_weight* assigned to the position in a term vector for which a term is present in the BOW. This can increase or decrease the importance of Word2vec similarity values that are assigned to positions in the vector for which the term is not present in the BOW. This balances the effect of exact term matches in relation to Word2vec similarity scores. Parameter (2) determines the minimum vector similarity (which we refer to as *min_vec_sim*). Next, we need to select the optimal W2V configuration upon which Word2vec is trained to calculate similarities for our “waste” domain, that is, comparing, the “Google News” configuration, the “Common Crawl” configuration, and the “Elsevier” configuration. Finally, we need to select the better of the two approaches proposed to represent the context-based term similarity created by the Word2vec similarity score resulting in the term vector configuration: *“Average W2V Similarity”* or *“Maximum W2V Similarity”*.

4.3.1. Parameter setting

To determine the optimal term weight, we test the sensitivity of *term_weight* for all recommenders using method C or D to detect where the best modified balanced accuracy is achieved. The *term_weight* determines whether the CosSim-based match (see Section 3) puts more emphasis on a term that exists in both the BOW $b_i$and b_t or on a term that exists only in one BOW; thus, a match is created using a representation of term similarity based on the Word2vec similarity score(s). Fig. 2 explains the initialization of the *term_weight* parameters. After some noise at 0–0.5, most of the recommenders show an increasing modified balanced accuracy with a decreasing slope between 0.5 and 2.0 followed by a slight decrease (2.0–5.0), resulting in a concave function with a maximum around term-weight equal to 2. Therefore, we selected 2 as the *term_weight* parameter value for our experiment, as this seems to provide close to the best results for all recommenders.

We also perform a sensitivity analysis with respect to the minimum vector similarity *min_vec_sim*. Two important aspects should be considered here. First, it concerns a multi-class recommender problem with many tags (each tag forms a class). In addition, a term match between two short-text descriptions cannot always be established (see Fig. 4; the number of list recommendations generated is lower than 311, which is the number of items for which recommendation is required). Hence, when the vector similarity is increased to favor better recommendations, it may, unfortunately, also cut off the number of correct recommendations. As can be observed in Fig. 3 the modified balanced accuracy drops almost immediately, (as opposed to recommender systems in which a multi-fold of recommendations can be correct). This explains the initialization of the minimum vector similarity parameter to a value of 0.08 (just before the decreasing slope of the first recommender).

4.3.2. Configuration selection

To support the selection of the Word2vec data configuration Fig. 5 shows the result of a sensitivity analysis between the number of recommendations caused by adjusting the *min_vec_sim* parameter (2) (see Fig. 4), and the effect on the modified balanced accuracy. This analysis highlights the three Word2vec configurations with different line styles and colors. The preferred setting for our recommender would be to provide a recommendation in as many scenarios as possible while sustaining a satisfactory modified balanced accuracy. Consistent with the previous modified balanced accuracy results, it seems that the best representations of phrases are learned by a model with the “Common Crawl” configuration (dotted lines), reflected by the higher accuracy results over the curve by all recommenders using this configuration.

The term vector configuration *“Maximum W2V Similarity”* is selected for two reasons. First, the lower sensitivity of the configuration provides a more reliable modified balanced accuracy (see Fig. 2). Second, in Fig. 6 the sensitivity analysis between the number of recommendations as a result of adjusting Parameter (2), the *min_vec_sim*, and the modified balanced accuracy shows that the *“Maximum W2V Similarity”*

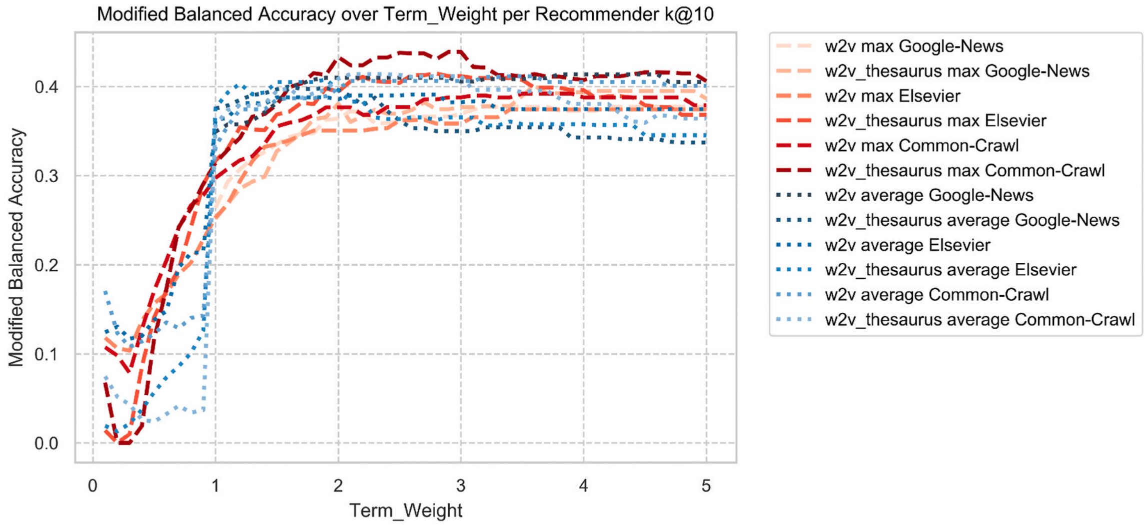

Fig. 2. Influence of parameter (1) *term_weight* on the modified balanced accuracy. Spearman’s Rank values range from ρ = −0.04 to 0.98 indicating that for some algorithms, there seems to be an positive underlying fitting monotonic relationship between the parameter variables ($\rho > 0.9$), but most algorithms are affected by the noise in [0–1] or the downwards slope in [2.0–5.0].

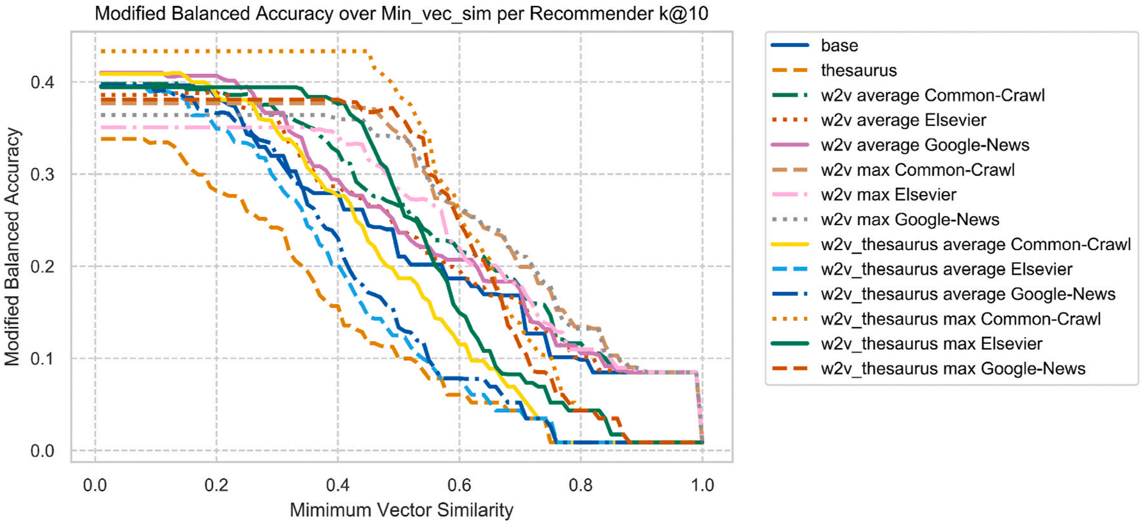

Fig. 3. Influence of parameter (2) "Minimum Vector similarity" on the modified balanced accuracy achieved by a recommender. Spearman's Rank values range from ρ = −0.94 to −0.99 indicating that for all algorithms, there seems to be an underlying fitting negative monotonic relationship between the parameter variables ($\rho > 0.9$).

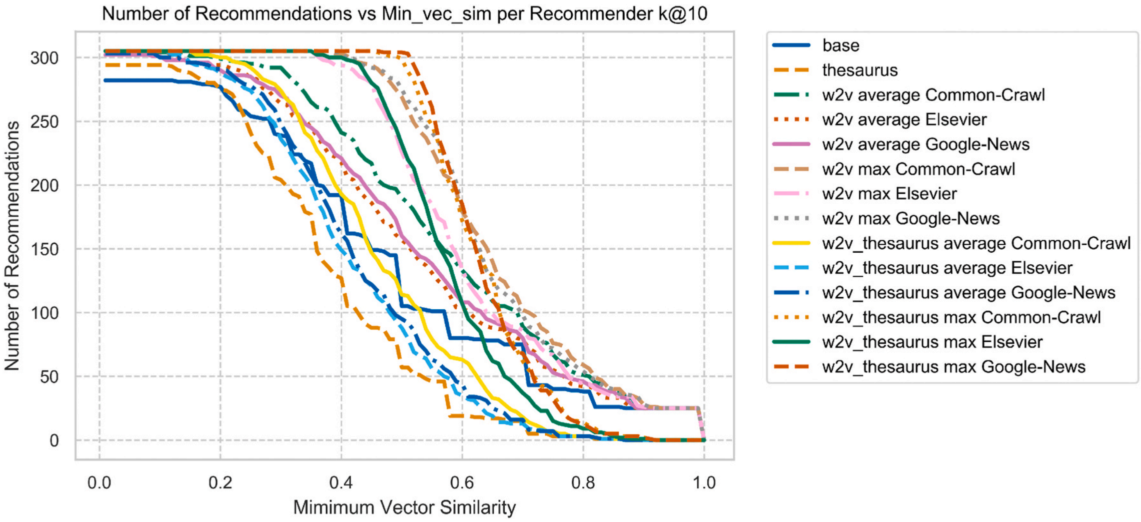

Fig. 4. Influence of parameter (2) "Minimum Vector similarity" on the number of recommendations generated by the recommender. Spearman's Rank values range from ρ = −0.95 to −0.99 indicating that for all algorithms, there seems to be an underlying fitting negative monotonic relationship between the parameter variables ($\rho > 0.9$).

configuration (dashed lines) mostly outperforms the modified balanced accuracy of the *"Average W2V Similarity"* configuration (dotted lines).

4.4. Comparison of methods

To compare the methods we use the optimal configuration as explained in Section 4.3. These are (A) base-line method, (B) thesaurus, (C) Word2vec in the configuration "Common Crawl" using the context-based term similarity "Maximum W2V Similarity", and (D) Word2vec + thesaurus in the configuration "Common Crawl" using the context-based term similarity "Maximum W2V Similarity". The prediction task was evaluated with $k = 10$ using a *min_vec_sim* of 0.08 and a *term_weight* of 2. Fig. 7 shows the performance of these four methods reporting the scores for each of the evaluation metrics defined in Section 4.2. The detailed data underlying Fig. 7 are presented in Table 8.

As it concerns an evaluation of recommender systems on a multi-class classification problem, we measured a recommendation (a list of tags) at the micro level, at the macro level and by using rank-based metrics. The micro and macro evaluation metrics (e.g., micro precision, modified balanced accuracy) measure at a list level, implying that if the correct tag is among the first *k* ranked tags, it is considered as a correct list of tags and thus a correct recommendation. These list-level metrics show a noticeable decrease in performance for recommender methods B and C, while the recommender method D shows an increase (all in comparison with method A). The micro results, that is, the micro precision, micro recall, micro accuracy, and micro F1 scores are all equal, as can be observed in Table 8. This can be explained as follows. First, a *fp*for tag *t* is also a *fn* for (correct) t', as denoted in Equation (20) where $t' \neq t$.

$$fp_t = fn_{t'} \tag{20}$$

Similarly, a *fn* for tag *t* is also a *fp* for tag t' (i.e., the tag that is presumed to be the correct tag).

$$fn_t = fp_{t'} \tag{21}$$

From this, it follows that the sum of all fp_t is equal to the sum of all fn_t

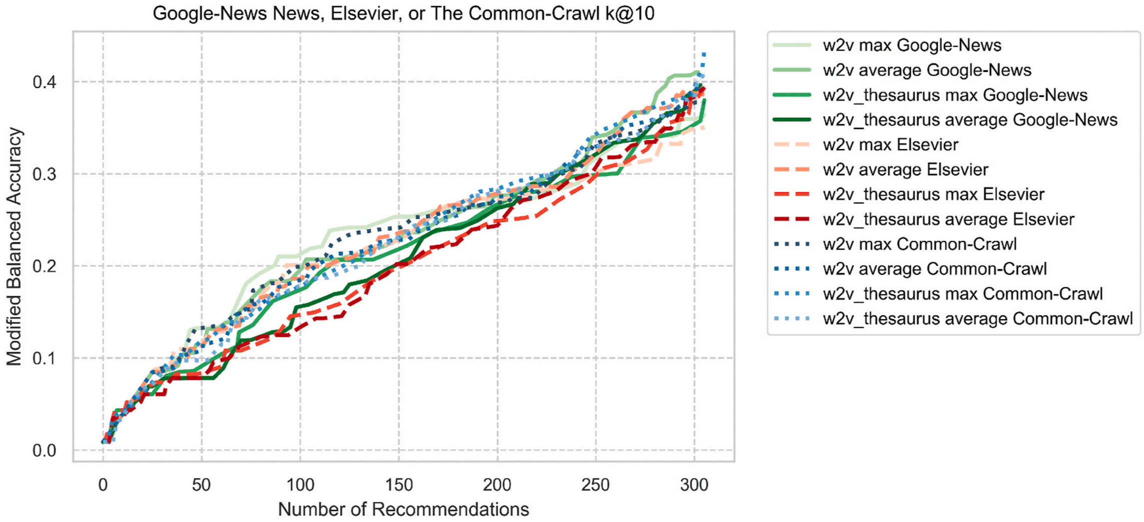

Fig. 5. Influence of different Word2vec configurations for learning word embeddings that affect the modified balanced accuracy of the recommenders. Spearman's Rank values range from ρ = 0.97 to 0.99 indicating that for all algorithms, there seems to be an underlying fitting positive monotonic relationship between the parameter variables ($\rho > 0.9$).

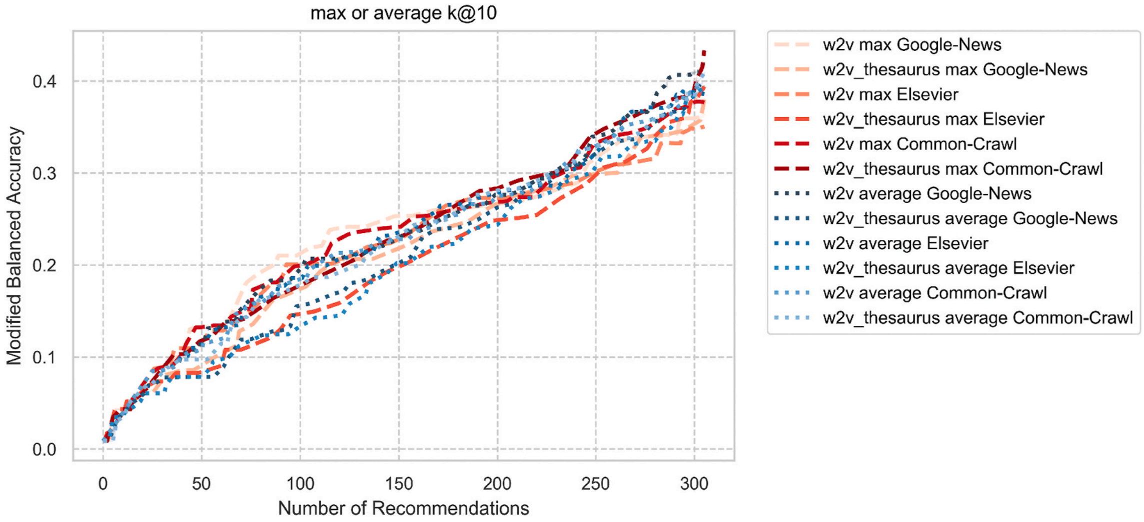

Fig. 6. Influence of term vector configuration "Average" or "Max" on the modified balanced accuracy of recommenders. This is a pivot of Fig. 5, hence identical ρ values.

(see Equation (22)), which gives identical results for $precision_\mu$, $recall_\mu$, $F - measure_\mu$, and $accuracy_\mu$.

$$\sum_{t\in T} fp_t = \sum_{t\in T} fn_t \tag{22}$$

On the other hand, the rank-based evaluation *DCG*, which assigns a score based on the position where a correct tag is found in the list, shows a decreasing rank for Method B, whereas the rank of Method C and D are almost similar to that of A. The *MRR* in our context assigns a higher score to a lower-ranked correct tag than *DCG*, as reflected in the *MRR* results that show a greater difference. Although an increase in micro precision for method D benefits the support that can be provided during the EWC classification task in online waste exchange platforms, the results are only marginal. Concerning the number of recommendations provided, we notice an increase (282–305) for methods using the thesaurus and Word2vec.

To understand the significance of the results, we apply a significance test. Our data are non-normally distributed, indicated by the Shapiro-Wilk (Shapiro and Wilk, 1965) test results that report *p*-values of < 2.2e-16 for all data. Therefore, to statistically compare the performance we use two non-parametric statistical significance tests (see Table 9). For the list-based metrics, we use McNemar's test (McNemar, 1947), which is a non-parametric test for paired nominal data. For the rank-based metrics, we use the Wilcoxon's signed rank test (Wilcoxon and Wilcox, 1964), which is a non-parametric test for paired ordinal and continuous data.

The Wilcoxon test compares the alternative versions of the proposed algorithms (B,C,D) to the baseline algorithm (A) based on the paired results (i.e., between binary evaluation values of recommendations) without making distributional assumptions on the differences (Shani and Gunawardana, 2011). Such a hypothesis is non-directional, and therefore the two-sided test is applied. We tested by using an α of 0.05, implying a 95% confidence interval. The resulting Wilcoxon Critical *T* values for each sample size *n* (reported in Table 9) were obtained from the *qsignrank* function in *R* (Dutang and Kiener, 2019). The Wilcoxon test is configured to adjust for the ranking in the case of ties. Furthermore, it is initialized with the setting that accounts for the likely binomial distribution in a small data set by applying continuity correction.

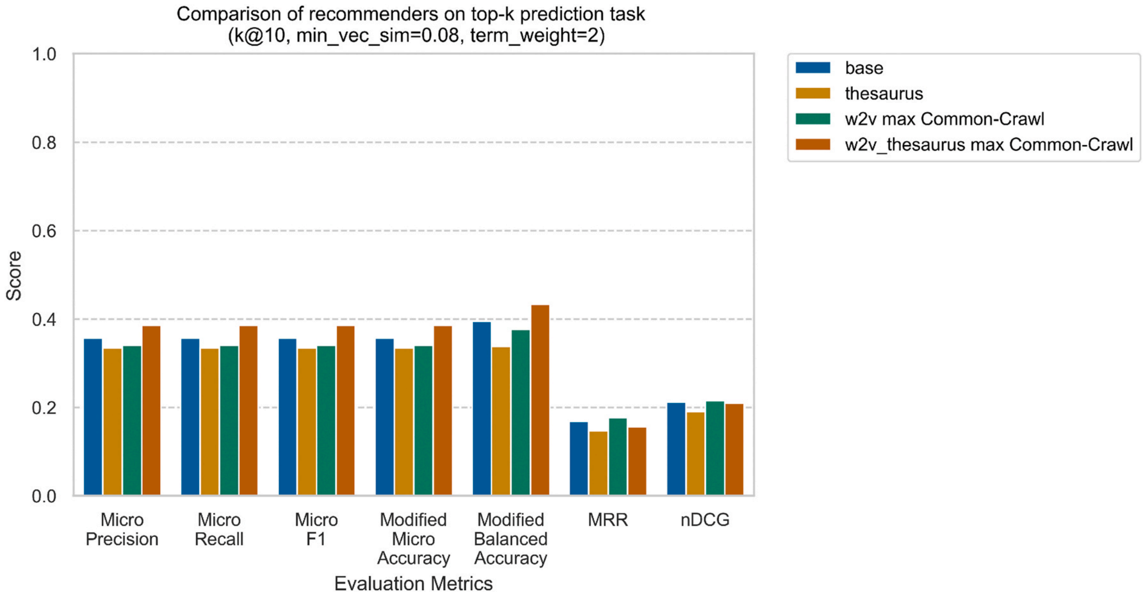

Fig. 7. Evaluation metrics.

Table 8
Comparison of recommenders on top-k tag prediction task. Method C and D use configuration “Common Crawl”, and method “Maximum Vector Similarity”, term_weight = 2, min_vec_sim = 0.08, evaluation k@10.

Method	#Recommended Items (list)	Micro Precision (list)	Micro Recall (list)	Micro F1 (list)	Modified Micro Accuracy (list)	Modified Balanced Accuracy (list)	MRR (rank)	DCG (rank)
A: baseline	282	0.3569	0.3569	0.3569	0.3569	0.3952	0.1683	0.2120
B: thesaurus	294	0.3344	0.3344	0.3344	0.3344	0.3378	0.1473	0.1906
C: Word2vec	305	0.3408	0.3408	0.3408	0.3408	0.3765	0.1769	0.2155
[l]D: Word2vec + thesaurus	305	0.3859	0.3859	0.3859	0.3859	0.4332	0.1563	0.2100

Table 9
Statistical significance of recommender comparison: McNemar test and Wilcoxon signed rank test.

Test	Compared Method	Metric Type	n	X^2	Critical T	T	α	p
McNemar two-sided	A-B	List	60	1.067	–	–	0.05	0.3017
McNemar two-sided	A-C	List	21	1.191	–	–	0.05	0.2752
McNemar two-sided	A-D	List	21	1.191	–	–	0.05	0.2752
Wilcoxon two-sided	A-B	Rank	117	–	2732	2724.5	0.05	0.0476
Wilcoxon two-sided	A-C	Rank	72	–	965	1093	0.05	0.2089
Wilcoxon two-sided	A-D	Rank	131	–	3471	4519	0.05	0.6518

As is evident from Table 9, only the test comparing Method A-B has a T lower than the Critical T; hence, the difference between Method A and Method B is significant at $p \leq 0.05$ (also reflected in the p-value of the test A-B, which is 0.0476 this value is less than the significance level $\alpha = 0.05$). We can conclude that the median weight of Method B is significantly different from the median weight of Method A. However, both the McNemar Test and the Wilcoxon rank test comparing Method A-C and Method A-D report p-values lower than the significance level $\alpha = 0.05$. However, for methods C and D $n < 30$ which indicates that there is only marginal performance change. McNemar’s test (McNemar, 1947) is typically valid when the number of discordant pairs exceeds 30 (Rotondi, 2019). Therefore, we cannot confirm or disprove the statistical validity of the list-based metrics for C and D. To investigate whether all the results are significant, a larger data set is required. A positive remark on the significance of the results is that combining a thesaurus with the contextualized Word2vec approach does not significantly worsen the ranking, but does increase the number of items that can be recommended. This is a positive characteristic that is helpful in content-based algorithms, which are built with the aim of bootstrap recommendation.

5. Discussion

5.1. *Causes for incorrect EWC tag recommendations*

To identify the causes behind the relatively low performance of the tag recommenders, we analyze how each of the ranked lists of EWC tags for a waste item is generated. The analysis considers all recommendations produced by the four compared recommenders (see Section 4.4), for all the 311 waste items. Table 10 provides a list of causes found, ranked by the frequency of observation, which explains why a filtering algorithm failed to retrieve the correct EWC tag. For an identical item, several causes may be registered for every recommendation generated by a different method. However, only one unique cause is counted for each item for which all recommenders attempt to suggest a tag.

Suggestions are provided in three areas to increase the performance of NLP-based EWC tag recommendation. First, the analysis of natural language may be improved by adding descriptions from the higher levels in the EWC hierarchy. These contain the originating industry specifics, which may help to distinguish between many similar types of EWC code

Table 10
Semantic challenges as reasons for why a filtering algorithm did not retrieve the correct EWC tag.

Code	Challenge Description	Count
C1	The description at level 1 or level 2 in the EWC hierarchy specifies the term required for term matching.	54
C2	The EWC tag description is “waste not other specified” or “other waste resulting from industry sector X”. This is inconclusive about what material it addresses.	42
C3	The thesaurus is incomplete and requires an update for a specific material (e.g., plastic to big bags, inert to concrete, glass to glazing). Some EWC tags use descriptions as “mixed waste”, where the thesaurus is expected to list the exact materials.	27
C4	A term match is realized on terms that are in general irrelevant. These could better be filtered or devalued (e.g., “use”, “bag”, “specify”, “product”, “treatment”, “contain”, “manufactur”, “organics”).	24
C5	The item description is vague, or too general (e.g., “home trash”, “solid waste”, “debris-demolition waste”).	19
C6	Terms with less than three characters are filtered. Therefore important materials were missed (e.g., oil).	17
C7	There are too many EWC tags addressing the same class of materials (e.g., there are codes for “general” metals in any forms, while there are also codes for specific metals such as iron). In the case of metals and plastics the thesaurus widens to an infinite number of options.	17
C8	The third level part of an EWC code uses the number 99 or a number equal to 1 plus the number of the EWC level-3 code of the highest classified leaf node. Typically this is used to indicate non-classifiable materials belonging to an EWC at level 2. However, the code 99 is also used in instances where there is no official EWC code ending at 99, thus these instances result in incorrectly classified items.	9
C9	The nouns (usually describing the material) are not prioritized over other terms. Adjectives or adverbs are perceived less important (e.g., water-based paint, hydrolic oil), the adjectives are a second specifier in case different EWC codes already contain that material noun.	9
C10	The item description does not mention the materials, but describes the class of materials (e.g., powder, slag, packaging), the state of the material (e.g., solid), and/or identifiers (contaminated, contains, hazardous).	8
C11	The item is tagged using a code indicating what the waste can become rather than where it originates from (e.g., sawmill dusts as paper packaging).	8
C12	There are two or more products in the description (e.g., there was metal and wood in one description).	8
C13	A potential mis-classification of the EWC tag. Either the tag was (1) wrong, or (2) there might be a better alternative tag.	7
C14	Sludge can contain particles (aluminum), the EWC tag refers to what can be extracted rather than what it is (there are also many sludge related EWC codes).	5
C15	A misspelling, or the use of a dialect form of a term (e.g., American and Canadian English (aluminum) vs British English (aluminium), misspellings “aerosolss” instead of “aerosoles”).	5
C16	The material was mentioned more than once, yet the algorithm focused on different terms.	4
C17	Some materials can result from different sectors, but in fact address the same materials; therefore, there are too many EWC tags for this material.	4
C18	The thesaurus is too broad, too many different types of material are listed (e.g., mixed waste lists all kinds of metals, plastic metal, wood, etc.).	4
C19	The adjective and noun form of a material are not recognized to be similar (wooden v.s. wood).	4
C20	Abbreviations are not captured (e.g., Waste PP-PE-PP).	3
C21	Term matching is focused on describing the classes of materials (e.g., powder) or states of the materials (e.g., solid).	2
C22	A material is addressed in two terms, which should be interpreted as one. (e.g., a big bag (which is a “Flexible intermediate bulk container”) or non-hazardous vs hazardous (negative specifier)).	2
C23	The product is meant for reuse, not for recycling. Therefore, the waste was listed with an EWC tag “Other”.	2
C24	The EWC tag is used to annotate the production process that produces the waste, while there exist tags to label the material	2
C25	Some relations are just difficult (e.g., to understand tank sludge hydro carbon is dixi waste and onsite effluent).	2
C26	One needs to differentiate between the production process terms and the material (e.g., glazed (process) tiles (material)).	1
C27	A non-significant term that was filtered in the pre-processing steps was found likely to be essential for creating a match.	1
C28	The link between an item description and the EWC description is the use of mentioned production equipment for certain materials.	1

descriptions. In addition, creating higher quality source data (e.g., improving the data cleaning of the natural language and enriching the thesaurus) could solve a number of problems from the list. Finally, designing a prioritization mechanism of important terms is considered to improve the performance of the recommender. For example, one might integrate a weighted terms approach in the Word2vec analysis, such as the classic term-frequency inverse document frequency. Another approach is to assign a weight to terms that determine a known material by deriving these from material databases.

The causes behind the poor functioning of the algorithms suggest a redesign of the proposed NLP-based filtering algorithm. However, careful consideration is necessary, as including new techniques in the filtering algorithm may, on the one hand, improve one set of recommendations, while simultaneously impairing other recommendations. A candidate identified is the design of a hybrid filtering algorithm based on a similarity match of natural language in combination with a support vector machine approach, learning the use of language associated with EWC codes. However, the prerequisite is a substantially larger number of classified EWC descriptions that can be used for training the algorithm.

5.2. EWC tag ontologies

The EWC system, as experienced in this study and supported by others (Sander et al., 2008), has its inefficiencies (e.g., hierarchical reporting structures), incapabilities (e.g., missing codes) and classification issues (for example, overlap between codes). In the current state, the system still suffers from the unsolved problems addressed in (Sander et al., 2008) and would benefit from a major update or redesign. Furthermore, we propose the need for a critical discussion on the use of EWC codes in all applications of waste registration, particularly, in tag recommendation. The initial design of the EWC was developed to facilitate statistical reporting. However, with developments and the use of the EWC in the waste domain, this is no longer the sole purpose of the EWC. Therefore, it could be argued that other types of structures for different purposes and another governing strategy may better apply waste item annotation.

There are three common approaches for governing and structuring tags. First, there is the use of an a priori defined tag ontology (also referred to as a “fixed” taxonomy (Nie et al., 2014; Gupta et al., 2010)) that restricts users to annotate items only with tags from this ontology. These are usually constructed by experts and have emerged because of the standardization of practices, or through legislation mandating the use of a particular taxonomy (Ciccarese et al., 2011). Furthermore, a set can be governed using folksonomy, which is typically found in social tagging systems. A folksonomy is an unstructured knowledge classification created to allow users to choose arbitrary tags for assignment to items (Dattolo et al., 2012). Finally, there is ontology governance through the extraction of concepts from a large text corpus using dimensionality reduction techniques. In these techniques, an artificial intelligence algorithm rather than an expert or common folk, is responsible for defining the latent ontology (Zhong et al., 2017). There are both positive and negative aspects of using a fixed taxonomy, a folksonomy, or a latent ontology approach. The advantages of a fixed taxonomy are that they are considered rigid, structured, conservative, and centralized, and work well in environments that require a standardized format that is consistent over time (Gupta et al., 2010). A folksonomy functions better than a fixed taxonomy when addressable concepts are time-dynamic or user-specific. Because folksonomy does not require a taxonomy custodian it may be managed in a decentralized manner (Nie et al., 2014). The latent ontology approach may be viewed as an evolved approach to govern a folksonomy, addressing the issue of overlap in concepts caused by the user-specific language (Zhong et al., 2017). This initiates a discussion on the development of a waste folksonomy instead of using the EWC.

6. Conclusion

There is a growing concern for the development of recommender systems to facilitate symbiotic development. This study addresses the problem of waste item classification using EWC tags. The contributions of this study are as follows: (i) we provide a performance comparison of recommender models that experiment with semantic enhancement (an EWC thesaurus) and the linguistic contexts of words (learned by Word2vec) in detecting term vector similarity in addition to direct term matching algorithms; (ii) we present an in-depth analysis providing insights into why the different recommenders were unable to generate a correct annotation that motivates a discussion on the current design of the EWC system.

The proposed models show that waste item annotation could be supported by recommenders; in addition, these technologies could also be adopted in a practical context. Our findings indicate several problems that need to be resolved before we achieve a fully optimized EWC tag recommender. To bootstrap the initial EWC tag recommendation, we focus our investigations on models that use short-text natural language processing. We propose a new model for tag recommendation that is based on term vector similarity matching derived from short-text descriptions and enriched with semantics and context. The results show that using both thesaurus and Word2vec could improve the accuracy of the term analogy task compared to the baseline term-matching model.

The three most prominent areas to improve are: (1) including upper-level descriptions in the EWC hierarchy for an EWC description, to distinguish between similar EWC codes that originate from different industries, (2) improving the data quality (for example, by filtering jargon-based stopwords and a revision of the thesaurus in general (which was observed to be incomplete)), and (3) designing a method for semantic weighting of terms (e.g., material-related terms, adjectives, and term-frequency).

Unfortunately, the domain of EWC recommendation is immature, and efforts are required to build a larger sample set to test tag recommendation algorithms. In view of these results, our work should be considered as a first attempt to guide the building of recommenders for the circular economy domain, wherein the EWC coding system plays a predominant role. Future studies may provide more substantial data support and test the generalizability of the claim that a combined effort of a thesaurus and Word2vec leads to improved accuracy in other domains. Furthermore, the numerous data problems mentioned in this paper provide directions for the improvement of algorithm design concerning EWC tag recommendation. These types of algorithms may be applied in industrial symbiosis platforms, public waste registers (e.g., the European Pollutant Release and Transfer Register (E-PRTR)), waste permit and certificate registers (e.g., those operated by the Environmental Protection Agencies), and other private systems, facilitating the daily operations of waste carriers, brokers, and dealers.

Author credit

Guido van Capelleveen – Main author. Chintan Amrit – Supervision, review & editing. Henk Zijm – Supervision, review & editing. Devrim Murat Yazan – Supervision, review & editing. Asad Abdi - Supervision, review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research is funded by European Union's Horizon 2020 program under grant agreement No. 680843.

References

Abdi, A., Idris, N., Alguliyev, R.M., Aliguliyev, R.M., 2015. Pdlk: plagiarism detection using linguistic knowledge. Expert Syst. Appl. 42, 8936–8946. https://doi.org/10.1016/j.eswa.2015.07.048.

Akther, A., Kim, H.-N., Rawashdeh, M., El Saddik, A., 2012. Applying latent semantic analysis to tag-based community recommendations. In: Kosseim, L., Inkpen, D. (Eds.), Advances in Artificial Intelligence. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 1–12.

Albitar, S., Fournier, S., Espinasse, B., 2014. An effective tf/idf-based text-to-text semantic similarity measure for text classification. In: Benatallah, B., Bestavros, A., Manolopoulos, Y., Vakali, A., Zhang, Y. (Eds.), Web Information Systems Engineering – WISE 2014. Springer International Publishing, Cham, pp. 105–114. https://doi.org/10.1007/978-3-319-11749-2_8.

Alepidou, Z.I., Vavliakis, K.N., Mitkas, P.A., 2011. A semantic tag recommendation framework for collaborative tagging systems. In: 2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing, pp. 633–636. https://doi.org/10.1109/PASSAT/SocialCom.2011.170.

Baeza-Yates, R., Ribeiro-Neto, B., 2011. Modern Information Retrieval: the Concepts and Technology behind Search, second ed. Addison-Wesley Publishing Company, USA.

Belém, F.M., Almeida, J.M., Gonçalves, M.A., 2017. A survey on tag recommendation methods. Journal of the Association for Information Science and Technology 68, 830–844. https://doi.org/10.1002/asi.23736.

Ben-Lhachemi, N., Nfaoui, E.H., 2017. An extended spreading activation technique for hashtag recommendation in microblogging platforms. In: Proceedings of the 7th International Conference on Web Intelligence, Mining and Semantics. WIMS '17, ACM, New York, NY, USA, p. 16. https://doi.org/10.1145/3102254.3102283, 1–16:8.

Bojanowski, P., Grave, E., Joulin, A., Mikolov, T., 2017. Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics 5, 135–146. https://doi.org/10.1162/tacl_a_00051.

Cantador, I., Konstas, I., Jose, J.M., 2011. Categorising social tags to improve folksonomy-based recommendations. Journal of Web Semantics 9, 1–15. https://doi.org/10.1016/j.websem.2010.10.001.

Chakrabarti, C., Luger, G.F., 2015. Artificial conversations for customer service chatter bots: architecture, algorithms, and evaluation metrics. Expert Syst. Appl. 42, 6878–6897. https://doi.org/10.1016/j.eswa.2015.04.067.

Chertow, M.R., 2000. Industrial symbiosis: literature and taxonomy. Annu. Rev. Energy Environ. 25, 313–337. https://doi.org/10.1146/annurev.energy.25.1.313.

Ciccarese, P., Ocana, M., Castro, L.J.G., Das, S., Clark, T., 2011. An open annotation ontology for science on web 3.0. J. Biomed. Semant. 2, S4. https://doi.org/10.1186/2041-1480-2-S2-S4.

Commission Decision on the European List of Waste, 2000. European Commission (COM 2000/532/EC), Technical Report, European Commission.

Common Crawl, 2019. Common crawl. Accessed: https://commoncrawl.org/. (Accessed 26 October 2019).

Crockett, K., McLean, D., O'Shea, J.D., Bandar, Z.A., Li, Y., 2006. Sentence similarity based on semantic nets and corpus statistics. IEEE Trans. Knowl. Data Eng. 18, 1138–1150. https://doi.org/10.1109/TKDE.2006.130.

Croft, D., Coupland, S., Shell, J., Brown, S., 2013. A fast and efficient semantic short text similarity metric. In: 2013 13th UK Workshop on Computational Intelligence. UKCI), pp. 221–227. https://doi.org/10.1109/UKCI.2013.6651309.

Dattolo, A., Ferrara, F., Tasso, C., 2012. On Social Semantic Relations for Recommending Tags and Resources Using Folksonomies. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 311–326. https://doi.org/10.1007/978-3-642-23187-2_20.

Deshpande, M., Karypis, G., 2004. Item-based top-n recommendation algorithms. ACM Trans. Inf. Syst. 22, 143–177. https://doi.org/10.1145/963770.963776.

Dutang, C., Kiener, P., 2019. Cran Task View: Probability Distributions. Accessed. https://cran.r-project.org/web/views/Distributions.html. (Accessed 29 December 2019).

Ekstrand, M.D., Riedl, J.T., Konstan, J.A., 2011. Collaborative filtering recommender systems. Foundations and Trends in Human-Computer Interaction 4, 81–173. https://doi.org/10.1561/1100000009.

Elsevier, 2019. Elsevier Scopus apis. Accessed: https://dev.elsevier.com/sc_apis.html. (Accessed October 2019).

Facebook Inc., 2019. Word vectors for 157 languages. Accessed. https://fasttext.cc/docs/en/crawl-vectors.html. (Accessed 26 October 2019).

Gao, J., He, Y., Wang, Y., Wang, X., Wang, J., Peng, G., Chu, X., 2019. Star: spatio-temporal taxonomy-aware tag recommendation for citizen complaints. In: Proceedings of the 28th ACM International Conference on Information and

Knowledge Management, CIKM '19. ACM, New York, NY, USA, pp. 1903–1912. https://doi.org/10.1145/3357384.3357894.

Gatzioura, A., Sànchez-Marrè, M., Gibert, K., 2019. A hybrid recommender system to improve circular economy in industrial symbiotic networks. Energies 12, 3546. https://doi.org/10.3390/en12183546.

Genc, O., van Capelleveen, G., Erdis, E., Yildiz, O., Yazan, D.M., 2019. A socio-ecological approach to improve industrial zones towards eco-industrial parks. J. Environ. Manag. 250, 109507. https://doi.org/10.1016/j.jenvman.2019.109507.

Gibert, K., Izquierdo, J., Sànchez-Marrè, M., Hamilton, S.H., Rodríguez-Roda, I., Holmes, G., 2018. Which method to use? an assessment of data mining methods in environmental data science. Environ. Model. Software 110, 3–27. https://doi.org/10.1016/j.envsoft.2018.09.021 (special Issue on Environmental Data Science and Decision Support: Applications in Climate Change and the Ecological Footprint).

Godoy, D., Corbellini, A., 2016. Folksonomy-based recommender systems: a state-of-the-art review. Int. J. Intell. Syst. 31, 314–346. https://doi.org/10.1002/int.21753.

Godoy, D., Rodriguez, G., Scavuzzo, F., 2014. Leveraging semantic similarity for folksonomy-based recommendation. IEEE Internet Computing 18, 48–55. https://doi.org/10.1109/MIC.2013.26.

Golder, S.A., Huberman, B.A., 2006. Usage patterns of collaborative tagging systems. J. Inf. Sci. 32, 198–208. https://doi.org/10.1177/0165551506062337.

Gomaa, W.H., Fahmy, A.A., 2013. A survey of text similarity approaches. Int. J. Comput. Appl. 68, 13–18. https://doi.org/10.5120/11638-7118, full text available.

Google, 2019. Google Code Archive: Word2vec. Accessed. https://code.google.com/archive/p/word2vec/. (Accessed 26 October 2019).

Grave, E., Bojanowski, P., Gupta, P., Joulin, A., Mikolov, T., 2018. Learning word vectors for 157 languages. In: Proceedings of the International Conference on Language Resources and Evaluation. LREC. http://arxiv.org/abs/1802.06893.

Gupta, M., Li, R., Yin, Z., Han, J., 2010. Survey on social tagging techniques. SIGKDD Explor. Newsl. 12, 58–72. https://doi.org/10.1145/1882471.1882480.

Hevner, A.R., March, S.T., Park, J., Ram, S., 2004. Design science in information systems research. MIS Q. 28, 75–105. https://doi.org/10.2307/25148625.

Hölbling, G., Thalhammer, A., Kosch, H., 2010. Content-based tag generation to enable a tag-based collaborative tv-recommendation system. In: Proceedings of the 8th European Conference on Interactive TV and Video. EuroITV '10, ACM, New York, NY, USA, pp. 273–282. https://doi.org/10.1145/1809777.1809832.

Hsu, I.-C., 2013. Integrating ontology technology with folksonomies for personalized social tag recommendation. Appl. Soft Comput. 13, 3745–3750. https://doi.org/10.1016/j.asoc.2013.03.004.

Huang, A., 2008. Similarity measures for text document clustering. In: Proceedings of the Sixth New Zealand Computer Science Research Student Conference (NZCSRSC2008). New Zealand, Christchurch, pp. 49–56.

Järvelin, K., Kekäläinen, J., 2002. Cumulated gain-based evaluation of ir techniques. ACM Trans. Inf. Syst. 20, 422–446. https://doi.org/10.1145/582415.582418.

Joulin, A., Grave, E., Bojanowski, P., Mikolov, T., 2016. Bag of Tricks for Efficient Text Classification, CoRR Abs/1607, 01759. http://arxiv.org/abs/1607.01759.

Khan, A., Baharudin, B., Lee, L.H., Khan, K., 2010. A review of machine learning algorithms for text-documents classification. J. Adv. Inf. Technol. 1, 4–20. https://doi.org/10.4304/jait.1.1.4-20.

Ko, Y., Park, J., Seo, J., 2004. Improving text categorization using the importance of sentences. Inf. Process. Manag. 40, 65–79. https://doi.org/10.1016/S0306-4573(02)00056-0.

Manning, C.D., Raghavan, P., Schütze, H., 2008. Introduction to Information Retrieval, vol. 1. Cambridge university press Cambridge.

McNemar, Q., 1947. Note on the sampling error of the difference between correlated proportions or percentages. Psychometrika 12, 153–157. https://doi.org/10.1007/BF02295996.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J., 2013. Distributed representations of words and phrases and their compositionality. In: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (Eds.), Advances in Neural Information Processing Systems 26. Curran Associates, Inc., pp. 3111–3119. http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf

Mirizzi, R., Ragone, A., Di Noia, T., Di Sciascio, E., 2010. Semantic tag cloud generation via dbpedia. In: Buccafurri, F., Semeraro, G. (Eds.), E-Commerce and Web Technologies. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 36–48.

Natural Language Tool Kit, 2019. Nltk 3.4.5 Documentation. Accessed. https://www.nltk.org/. (Accessed 26 October 2019).

Nie, L., Zhao, Y.-L., Wang, X., Shen, J., Chua, T.-S., 2014. Learning to recommend descriptive tags for questions in social forums. ACM Trans. Inf. Syst. 32 https://doi.org/10.1145/2559157, 5:1–5:23.

O'Shea, J., Bandar, Z., Crockett, K., McLean, D., 2008. A comparative study of two short text semantic similarity measures. In: Nguyen, N.T., Jo, G.S., Howlett, R.J., Jain, L.C. (Eds.), Agent and Multi-Agent Systems: Technologies and Applications. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 172–181.

Partalas, I., Kosmopoulos, A., Baskiotis, N., Artières, T., Paliouras, G., Gaussier, É., Androutsopoulos, I., Amini, M., Gallinari, P., 2015. LSHTC: A Benchmark for Large-Scale Text Classification, CoRR Abs/1503, 08581. http://arxiv.org/abs/1503.08581.

Patel, K., Bhattacharyya, P., 2017. Towards lower bounds on number of dimensions for word embeddings. In: Proceedings of the Eighth International Joint Conference on Natural Language Processing, ume 2. Asian Federation of Natural Language Processing, Taipei, Taiwan, pp. 31–36. Short Papers. https://www.aclweb.org/anthology/I17-2006.

Pedersen, T., 2008. Computational Approaches to Measuring the Similarity of Short Contexts : A Review of Applications and Methods, CoRR Abs/0806, p. 3787. http://arxiv.org/abs/0806.3787.

Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S., 2007. A design science research methodology for information systems research. J. Manag. Inf. Syst. 24, 45–77. https://doi.org/10.2753/MIS0742-1222240302.

Pennington, J., Socher, R., Manning, C., 2014. Glove: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, Doha, Qatar, pp. 1532–1543. https://doi.org/10.3115/v1/D14-1162.

Pires, A., Martinho, G., Chang, N.-B., 2011. Solid waste management in european countries: a review of systems analysis techniques. J. Environ. Manag. 92, 1033–1050. https://doi.org/10.1016/j.jenvman.2010.11.024.

Porter, M.F., 1980. An algorithm for suffix stripping, Program 14, 130–137.

Qassimi, S., Abdelwahed, E.H., Hafidi, M., Lamrani, R., 2016. Enrichment of ontology by exploiting collaborative tagging systems: a contextual semantic approach. In: 2016 Third International Conference on Systems of Collaboration. SysCo, pp. 1–6. https://doi.org/10.1109/SYSCO.2016.7831337.

Ratcliff, J.W., Metzener, D.E., 1988. Pattern-matching-the gestalt approach. Dr. Dobb's J. 13, 46.

Ribeiro, I.S., Santos, R.L., Gonçalves, M.A., Laender, A.H., 2015. On tag recommendation for expertise profiling: a case study in the scientific domain. WSDM '15, ACM, New York, NY, USA, pp. 189–198. https://doi.org/10.1145/2684822.2685320. Proceedings of the Eighth ACM International Conference on Web Search and Data Mining.

Rotondi, M.A., 2019. R Documentation: Mcnemar Pair-Matched Analysis Tool. Accessed. https://www.rdocumentation.org/packages/epibasix/versions/1.5/topics/mcNemar. (Accessed 2 January 2020).

Said, A., Bellogin, A., 2014. Comparative recommender system evaluation: benchmarking recommendation frameworks. In: Proceedings of the 8th ACM Conference on Recommender Systems, RecSys '14, Foster City, Silicon Valley, California, USA. ACM, New York, NY, USA, pp. 129–136. https://doi.org/10.1145/2645710.2645746.

Said, A., Bellogín, A., 2018. Recommender Systems Evaluation. Springer New York, New York, NY, pp. 2095–2112.

Sander, K., Schilling, S., Lüskow, H., Gonser, J., Schwedtje, A., Küchen, V., 2008. Review of the European List of Waste. Technical Report, Ökopol GmbH and ARGUS GmbH.

Scikit-learn developers, 2020. Metrics and Scoring: Quantifying the Quality of Predictions. Scikit-learn.org. https://scikit-learn.org/stable/modules/model_evaluation.html#balanced-accuracy-score.

Shah, R., Zimmermann, R., 2017. Tag Recommendation and Ranking. Springer International Publishing, Cham, pp. 101–138. https://doi.org/10.1007/978-3-319-61807-4_4.

Shani, G., Gunawardana, A., 2011. Evaluating Recommendation Systems. Springer US, Boston, MA, pp. 257–297. https://doi.org/10.1007/978-0-387-85820-3_8.

Shapiro, S.S., Wilk, M.B., 1965. An analysis of variance test for normality (complete samples). Biometrika 52, 591–611. http://www.jstor.org/stable/2333709.

Sharebox Project, 2017. Horizon2020 Spire-06-2015 Energy and Resource Management Systems for Improved Efficiency in the Process Industries: Project Sharebox. grant agreement no. 680843. unpublished workshop data). https://www.sharebox-project.eu/.

Sokolova, M., Lapalme, G., 2009. A systematic analysis of performance measures for classification tasks. Inf. Process. Manag. 45, 427–437. https://doi.org/10.1016/j.ipm.2009.03.002.

Sriram, B., Fuhry, D., Demir, E., Ferhatosmanoglu, H., Demirbas, M., 2010. Short text classification in twitter to improve information filtering. In: Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR '10, ACM, New York, NY, USA, pp. 841–842. https://doi.org/10.1145/1835449.1835643.

Subramaniyaswamy, V., Chenthur Pandian, S., 2012. Effective tag recommendation system based on topic ontology using wikipedia and wordnet. Int. J. Intell. Syst. 27, 1034–1048. https://doi.org/10.1002/int.21560.

Subramaniyaswamy, V., Vijayakumar, V., Indragandhi, V., 2013. A review of ontology-based tag recommendation approaches. Int. J. Intell. Syst. 28, 1054–1071.

Toderici, G., Aradhye, H., PasÃ§a, M., Sbaiz, L., Yagnik, J., 2010. Finding meaning on youtube: tag recommendation and category discovery. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 3447–3454. https://doi.org/10.1109/CVPR.2010.5539985.

U.K. Environment Agency, 2006. Using the List of Wastes to Code Waste: for Waste Transfer Notes, PPC Permits and Waste Management Licences in England & Wales. U.K. Environment Agency. Technical Report.

van Capelleveen, G., Amrit, C., Yazan, D.M., Zijm, H., 2018. The Influence of Knowledge in the Design of a Recommender System to Facilitate Industrial Symbiosis Markets. Environmental Modelling & Software. https://doi.org/10.1016/j.envsoft.2018.04.004.

van Capelleveen, G., Amrit, C., Yazan, D.M., Zijm, H., 2019. The recommender canvas: a model for developing and documenting recommender system design. Expert Syst. Appl. 129, 97–117. https://doi.org/10.1016/j.eswa.2019.04.001.

Wang, Y., Varadharajan, V., 2007. Role-based recommendation and trust evaluation. In: The 9th IEEE International Conference on E-Commerce Technology and the 4th IEEE International Conference on Enterprise Computing. E-Commerce and E-Services (CEC-EEE 2007), pp. 278–288. https://doi.org/10.1109/CEC-EEE.2007.83.

Wetzker, R., Zimmermann, C., Bauckhage, C., Albayrak, S., 2010. I tag, you tag: translating tags for advanced user models. In: Proceedings of the Third ACM International Conference on Web Search and Data Mining. WSDM '10, ACM, New York, NY, USA, pp. 71–80. https://doi.org/10.1145/1718487.1718497.

Wilcoxon, F., Wilcox, R.A., 1964. Some Rapid Approximate Statistical Procedures. Lederle Laboratories.

Zhang, C.-J., Zeng, A., 2012. Behavior patterns of online users and the effect on information filtering. Phys. Stat. Mech. Appl. 391, 1822–1830. https://doi.org/10.1016/j.physa.2011.09.038.

Zhao, S., Zhou, M.X., Yuan, Q., Zhang, X., Zheng, W., Fu, R., 2010. Who is talking about what: social map-based recommendation for content-centric social websites. ACM, New York, NY, USA, pp. 143–150. https://doi.org/10.1145/1864708.1864737. Proceedings of the Fourth ACM Conference on Recommender Systems, RecSys '10.

Zheng, A., Casari, A., 2018. Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists, first ed. O'Reilly Media, Inc.

Zhong, S., Lei, K., Huang, X., Wu, J., 2017. Topic representation: a novel method of tag recommendation for text. In: 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA), pp. 671–676. https://doi.org/10.1109/ICBDA.2017.8078720.

Řehůřek, Radim, 2019. Gensim: Topic Modeling for Humans. Accessed. https://radimrehurek.com/gensim/. (Accessed 26 October 2019).