Few-shot entity linking of food names

DOI:
10.1016/j.ipm.2023.103463

Document Version
Final published version

Information Processing and Management 60 (2023) 103463

Contents lists available at ScienceDirect

Information Processing and Management

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

Few-shot entity linking of food names

Darius Feher [a], Faridz Ibrahim [a], Zhuyan Cheng [a], Viktor Schlegel [a,c], Tom Maidment [b], James Bagshaw [b], Riza Batista-Navarro [a,*]

[a] University of Manchester, Oxford Road, Manchester, M13 9PL, United Kingdom
[b] E.Mission Innovation Ltd, 26 Westwood Road, Coventry, CV5 6GE, United Kingdom
[c] ASUS Intelligent Cloud Services, 37 Craig Rd, #02-01, Singapore

ARTICLE INFO

Keywords:
Entity linking
Natural language processing
Machine learning
Food knowledge base

ABSTRACT

Entity linking (EL), the task of automatically matching mentions in text to concepts in a target knowledge base, remains under-explored when it comes to the food domain, despite its many potential applications, e.g., finding the nutritional value of ingredients in databases. In this paper, we describe the creation of new resources supporting the development of EL methods applied to the food domain: the E.Care Knowledge Base (E.Care KB) which contains 664 food concepts and the E.Care dataset, a corpus of 468 cooking recipes where ingredient names have been manually linked to corresponding concepts in the E.Care KB. We developed and evaluated different methods for EL, namely, deep learning-based approaches underpinned by Siamese networks trained under a few-shot learning setting, traditional machine learning-based approaches underpinned by support vector machines (SVMs) and unsupervised approaches based on string matching algorithms. Combining the strengths of each of these approaches, we built a hybrid model for food EL that balances the trade-offs between performance and inference speed. Specifically, our hybrid model obtains 89.40% accuracy and links mentions at an average speed of 0.24 seconds per mention, whereas our best deep learning-based model, SVM model and unsupervised model obtain accuracies of 86.99%, 87.19% and 87.43% at inference speeds of 0.007, 0.66 and 0.02 seconds per mention, respectively.

1. Introduction and background

In order to achieve the United Nations Sustainable Development Goals, food systems need to be transformed to deliver healthier diets, support environmental sustainability and make food accessible to everyone, especially to those who are already suffering from food insecurity (Fears et al., 2019). In 2019, the EAT Lancet Commission proposed strategies towards a Great Food Transformation, highlighting the importance of good data on diets and food systems (Willett et al., 2019).

Data on food comes in either structured or unstructured form. On the one hand, there exist various databases developed by authoritative bodies containing *structured information* on food, that focus on nutrition or composition (Harrington et al., 2019), for example. Results of diet or nutrition studies (e.g., from surveys) are also often collected in a structured form (Miller et al., 2021). On the other hand, cooking recipes – which consumers tend to engage with on a regular basis (Morning Consult, 2022) – are written in natural language and thus contain *unstructured information*. Given that consumers consult recipe websites to help them decide on what food to prepare and eat, recipes play an important role in the food selection and consumption process (Silva et al., 2019). They also have the potential to facilitate dietary analysis; for instance, nutrition data on specific ingredients in a recipe can be retrieved

* Corresponding author.
E-mail address: riza.batista@manchester.ac.uk (R. Batista-Navarro).

https://doi.org/10.1016/j.ipm.2023.103463
Received 11 March 2023; Received in revised form 5 July 2023; Accepted 11 July 2023
Available online 26 July 2023

from a food nutrition database to estimate the number of calories in the resulting dish. However, as recipes are contributed by authors who vary in terms of cultural backgrounds, writing styles and language use, matching ingredient names in recipes with their corresponding canonical names in databases poses a technical challenge.

The process of automatically matching mentions (e.g., food names) in text, to entries or concepts in a target knowledge base is a natural language processing (NLP) task known as entity linking (EL). A popular EL problem, for example, is the linking of mentions (e.g., in web pages or documents) to Wikipedia entries – a task known as Wikification (Moro et al., 2014). In this paper, we focus on linking food names appearing in any given recipe's ingredient list to their corresponding canonical names in a food knowledge base. Assume, for example, that the said food knowledge base contains an entry for 'Aubergine' which pertains to the purple, absorbent vegetable often used in cooking. Given the mention 'eggplants' in a recipe, an EL solution should identify 'Aubergine' as the best-matching concept.

We now provide a more formal definition of EL, which is also known in the literature as named entity disambiguation (NED), entity normalisation, entity grounding, or entity categorisation. Given a knowledge base containing a set of concepts C and a text document in which a set of mentions M have been identified, an EL solution maps each $m \in M$ to the corresponding concept $c \in C$ in the target knowledge base. A *mention* is a text span of interest, which may consist of one or more words (i.e., tokens). In the context of this work, mentions pertain to food names, e.g., 'salt' or 'extra virgin olive oil'.

EL comes with a number of challenges. Some of them lie in *name variations*, whereby the same concept can be referred to in many ways in practice (e.g., 'Aubergine' can be referred to as 'eggplant' or 'aubergine' in recipes, depending on whether the recipe authors had used American or British English). There are also cases of *accidental lexical similarity*, in which multiple names appear to be lexically similar even if they refer to different, unrelated concepts (e.g., 'tamari', a food product that is similar to 'Soy sauce', bears no semantic relationship to the pod-like fruit 'Tamarind'). Another challenge is *ambiguity*, whereby a name could potentially refer to multiple concepts (e.g., 'pepper' could refer to 'Red pepper', 'Black pepper' or 'Sichuan pepper'). The above-mentioned challenges are aggravated in EL for food names in cooking recipes, where ingredients are presented as a list of short textual descriptions that lack contextual information that can help in disambiguation (Wu et al., 2018). As knowledge bases tend to contain hundreds if not thousands of concepts, EL also requires careful consideration of *scalability and speed* – especially during inference – to ensure that the best-matching concept can be identified within reasonable time despite a large search space (Parravicini et al., 2019; Laskar et al., 2022).

In this paper, we investigate and quantitatively evaluate various approaches to EL in the food domain. The work presented here is part of a bigger project called Emissions Calculator for Recipes (E.Care), which was aimed at estimating the carbon footprint of food within the UK context. Nevertheless, the EL models that we developed can be utilised by other researchers: (1) those who might be similarly exploring EL solutions in their own domains of interest, and (2) researchers who are exploring other applications in the food domain, e.g., linking food names to supermarket databases to enable the automatic finding and purchasing of ingredients.[1]

In the remainder of this paper, we first provide a review of related work (Section 2). Informed by insights from this review, we present our overarching aim and research questions (Section 3). This is followed by a description of our methods for constructing a new food knowledge base and recipe dataset, which led to creation of the E.Care KB and the E.Care dataset, both of which support the development and evaluation of EL methods (Section 4). Importantly, we present details of the various EL approaches that we developed (Section 5) and the results of evaluating and combining them into one hybrid model (Section 6). We then analyse our results and discuss their implications (Section 7) before providing a summary of our findings as well as potential directions for future work (Section 8).

2. Related work

In this section, we provide an overview of previously reported related work. First, we present a summary of the computational tools and resources that have been made available to support the automated (or semi-automated) analysis of food-related information within text. This is followed by a description of the various automated tasks – both NLP and machine learning-based – that have been applied to the food domain. Importantly, we provide a comprehensive review of EL approaches, covering the state-of-the-art methods in both the food and the general domain.

2.1. Computational resources supporting NLP in the food domain

The food domain has attracted the attention of the NLP community in recent years, as textual data containing information relating to food became increasingly available. For instance, the Recipe Flow Graph (r-FG) corpus was constructed by Yamakata et al. (2020) to support the automatic generation of flow graphs based on instructional text. It consists of 200 English recipes sampled from the Allrecipes website.[2] Apart from names of ingredients, other types of mentions within text were also annotated including names of cooking tools, duration values, quantities, actions, and the state of ingredients and tools. Importantly, the relationships between these mentions were annotated to allow for the construction of flow graphs. Another corpus is FoodBase (Popovski et al., 2019c), which consists of recipes that were also sourced from Allrecipes, but were annotated in a different way. The food names in each

[1] As in the case of "shoppable recipes"; see https://chicory.co/shoppable-recipes.

[2] https://www.allrecipes.com/

recipe were automatically recognised using a rule-based method (Popovski et al., 2019a); these were then assigned semantic tags based on a taxonomy of thematic categories related to food (University of Glasgow, 2015). It is worth noting that the semantic tagging applied to the food names is different from EL. Tagging is based on identifying any number of thematic categories (in the taxonomy) that are relevant to the meaning conveyed by a given name; for example, 'grilled chicken' is tagged with two concepts: `Cooking` and `Fowls`. Meanwhile, EL aims to identify only one best-matching concept.

Although most of the food-related corpora consist of recipes, some datasets are composed of other types of documents. The POMELO corpus, for instance, is based on MEDLINE paper titles and abstracts, whereby food names were annotated (but not linked to concepts in a knowledge base or taxonomy) with a view to capturing interactions between food items and drugs (Hamon et al., 2017). These available corpora (or datasets) were constructed to support the development of methods for tasks such as named entity recognition (NER), semantic tagging and flow graph construction. However, none of them contains the kind of annotations required in the development and evaluation of food EL methods. In contrast, the E.Care dataset that we have developed as part of this work consists of recipes whereby food mentions have been linked to the best-matching concept in a knowledge base, to facilitate the training or evaluation of food EL solutions.

A number of ontological resources have been used as the knowledge base of concepts for normalising or disambiguating food names (Popovski et al., 2019b). These include vocabularies and ontologies developed specifically for the food domain, including: (1) the Open Food Facts Ontology (LIRMM, 2013), which underpins the Open Food Facts platform;[3] the Food Product Ontology (ITMO University, 2016), which allows manufacturers and regulators to publish data on food products; and FoodOn (Dooley et al., 2018), which consolidates concepts from a number of food-related ontologies in the Open Biological and Biomedical Ontologies Foundry (OBO Foundry),[4] to represent the many different aspects of food (e.g., agriculture, harvested material, food products, consumption). None of the above-mentioned ontological resources were developed to cater to the UK context specifically, prompting us to develop our own E.Care Knowledge Base which contains information on food items relevant to the UK population.

2.2. Automated tasks relevant to the food domain

A number of machine learning-based and NLP tasks have proven to be applicable to the food domain. First is text classification: the task of categorising a piece of text according to predefined classes (labels). For instance, Mohammadi et al. (2020) developed classifiers for categorising recipes into four levels of difficulty (very easy, easy, fairly difficult and difficult) based on deep learning-based approaches such as convolutional neural networks (CNNs), gated recurrent units (GRUs), long short-term memory (LSTM) and transformer encoders. Different features such as pre-trained fastText embeddings (Bojanowski et al., 2017), bidirectional encoder representations from transformers (BERT) (Devlin et al., 2018) and CamemBERT embeddings (Martin et al., 2020) were employed to represent the recipes. In another work, Ma et al. (2022) framed the problem of categorising food based on ingredient statements as a multi-class classification task. They developed and compared traditional machine learning-based approaches, e.g., support vector machines (SVMs), with deep-learning-based ones (e.g., multi-layer perceptrons, recurrent neural networks). Bag of Words (BoW) and term frequency–inverse document frequency (TF–IDF) were used as feature representations.

Another task that has been explored within the food domain is NER. Popovski et al. (2019a) built FoodIE, a rule-based NER model that was developed without any reliance on the availability of annotated data. Stojanov et al. (2021) proposed FoodNER, which was based on fine-tuning BERT for the NER task. Similarly, MenuNER, developed by Syed et al. (2021), was underpinned by Bi-LSTM and conditional random field (CRF) models fed with BERT embeddings. Different from FoodNER and MenuNER, which were trained on restaurant reviews, the model proposed by Cenikj et al. (2020), BuTTER, was trained on labelled recipes in the FoodBase corpus (Popovski et al., 2019c). Similar to MenuNER, BuTTER's network architecture is based on Bi-LSTM and CRF layers built on top of BERT, but it makes use of character embeddings.

Clustering, a task which aims to find groups of similar instances in the data, has also been applied to analyse recipes. Ninomiya and Ozaki (2020) proposed a method to cluster recipes into four main dishes by representing them in two ways: one based on cooking steps in the recipe text using BERT and the other based on sequences of associated images (taken during the cooking process) using the CNN-based image recognition model VGG16. Another application of clustering was presented by Ventirozos et al. (2021); this used the HDBScan algorithm, in combination with a BERT-based model, to cluster embedding representations of recipe instructions. The resulting clusters correspond to different types of events that involve kitchen devices.

Although text classification, NER and clustering are tasks that are different from entity EL, the work reviewed above provided us with inspiration on the types of features suitable for representing recipe text. These include, e.g., BoW, fastText embeddings and BERT embeddings, which we similarly used in our own approaches (as will be described in Section 5).

2.3. Entity linking in the food domain

According to Popovski et al. (2019b), food EL is still an open research question. Indeed, at the time of writing, the food domain has thus far been under-explored when it comes to EL research. Only a handful of efforts in food EL have been reported, which we review below.

[3] https://world.openfoodfacts.org/

[4] https://obofoundry.org/

Eftimov et al. (2017) developed StandFood, a semi-automatic system that is capable of retrieving descriptions of food items through EL. Their approach is based on an unsupervised model that makes use of the Jaccard index and part-of-speech (POS) tagging to rank candidate concepts. However, the linking of mentions is performed by considering only lexical features (i.e., string similarity), and hence disregards any semantic information (i.e., semantic relationships between words). Chong and Lim (2018) developed an EL system for linking food-related social media posts to food concepts in a knowledge base, using an enhanced Bayesian model. Their system can be considered an implicit EL approach since it does not require the identification of food mentions in text as a prerequisite step. Instead, their model performs disambiguation by exploiting contextual information contained in an entire post. Popovski et al. (2019b) built the FoodOntoMap dataset which contains automatically generated links between food mentions in a corpus of recipes and their corresponding concepts in three ontologies, as well as mappings across the food concepts in those ontologies. Their EL approach is based on an unsupervised model that analyses semantic tags provided by the NCBO Annotator (Jonquet et al., 2009).

The lack of studies that investigated other approaches to food EL represents a research gap, one that we are seeking to address in our work by developing and comparing different food EL approaches.

2.4. Entity linking in the general domain

Outside of the food domain, a number of approaches for EL have been proposed, for example in biomedicine (Zheng et al., 2015; Karadeniz and Özgür, 2019; Yuan et al., 2022; Zhang et al., 2022; Chakraborty et al., 2023), news (Shanaz et al., 2021; Papantoniou et al., 2021) and social media (Basaldella et al., 2020). There are three broad types of EL approaches: unsupervised, traditional machine learning-based and deep learning-based.

2.4.1. Unsupervised approaches

Many unsupervised EL approaches employed dictionary look-up and string matching algorithms to link named entities to their corresponding concepts in a knowledge base (Wang et al., 2017; El Vaigh et al., 2020; Klie et al., 2020). For the purpose of normalising job titles, Spitters et al. (2010) compared different string similarity algorithms such as the token-based Jaccard, Dice, and Cosine similarity coefficients, and the character-based Q-grams, Levenshtein, Jaro, Jaro–Winkler and Needleman–Wunch indices. In their work, the Jaro and Jaro–Winkler indices were shown to obtain optimal performance. Meanwhile, Recchia and Louwerse (2013) performed toponym matching (i.e., linking names referring to the same place) based on multiple string matching methods and found that those based on the Smith–Waterman algorithm and longest common subsequence (LCS) performed best. Zheng et al. (2015) developed an unsupervised EL model for the biomedical domain that made use of Jaccard similarity and outperformed the state-of-the-art supervised model (at that time) by 9% in terms of accuracy. It is, however, worth noting that the performance of string matching algorithms is task-dependent (Recchia and Louwerse, 2013). As their performance in one domain cannot be generalised to other domains, we compare different string matching algorithms as part of our unsupervised approach to food EL (described in Section 5.4).

None of the above approaches took semantic similarity into consideration; therefore methods that measure similarity between names or mentions based on their word embedding representations have been proposed by Karadeniz and Özgür (2019) and Nozza et al. (2019). Such methods are also considered to be unsupervised, since training word embedding models does not require any labelled textual data.

2.4.2. Traditional machine learning-based approaches

Since a knowledge base can store a large number of concepts, casting EL as a multi-class classification task – whereby each concept is considered as a class – has become prohibitively expensive as it requires a classifier to learn to discriminate between hundreds or sometimes, even millions of classes (Neculoiu et al., 2016). Thus, a number of solutions formulate the EL problem as a binary classification task: given a pair consisting of a mention and a concept, the classifier infers whether the mention refers to the given concept or not. Some solutions employed traditional machine learning-based methods in developing their binary classification models, including logistic regression (El Vaigh et al., 2019) and Naïve Bayes (João et al., 2019). However, the majority of them made use of SVMs trained on features such as TF–IDF (Tsai et al., 2016; Alokaili and Menai, 2020). Thus we also built upon SVMs in developing our traditional machine-learning based approach (Section 5.3).

In cases where a binary classifier labels more than one pair as positive, multiple concepts would be returned as candidate matches for a given mention. Hence, techniques based on confidence scores, vector space models and SVM ranking were proposed for selecting the best-matching concept among multiple candidates (Hosseini et al., 2019). Meanwhile, other approaches directly rank concepts that have been identified as candidates, casting the EL task as a learning-to-rank (LTR) problem and employing algorithms such as SVMs, ListNet or LambdaMART (Zhang et al., 2011; Ceccarelli et al., 2013; Irrera and Silvello, 2021; Hosseini et al., 2021). These models utilise a variety of feature engineering techniques: term-based (e.g., frequency, syntactic or semantic), statistics-based, neural embedding-based (Hosseini et al., 2021) or graph-based (Hosseini et al., 2021; Irrera and Silvello, 2021). However, a downside of the LTR approach is its reliance on a dataset containing items that have been labelled based on their ranking. Such a dataset is expensive to construct especially if there is a large number of concepts of interest, as they need to be ranked with respect to their similarity to any given mention.

2.4.3. Deep learning-based approaches

On many NLP tasks, including EL, deep learning-based approaches have demonstrated state-of-the-art performance (Torfi et al., 2020), mostly owing to the emergence of transformer models: neural networks underpinned by the self-attention mechanism, which facilitates the learning of a contextual embedding representation of a sequence of tokens based on relationships between the tokens (Vaswani et al., 2017). Zhang et al. (2021) developed EntQA, a BERT-based approach that formulated EL as a question answering task, achieving state-of-the-art performance on a general-domain dataset (AIDA-CoNLL), with a micro-averaged F1-score of 85.8%. Another formulation of the EL task is that of De Cao et al. (2020) who cast EL as an auto-regressive text generation task, whereby a model learns to generate the concept based on a sentence that contains the mention of interest. Their approach obtained state-of-the-art performance according to evaluation on three general-domain datasets, namely, AIDA-YAGO2, WNED-CWEB and WNED-WIKI, where accuracy scores of 89.85%, 71.22% and 87.44% were obtained respectively.

There are other deep learning-based EL methods that formulated the problem as a classification task. An example of this is CHOLAN (Ravi et al., 2021), which employed a transformer model (powered by BERT) to encode information on each candidate concept based on its Wikipedia description and the context in which the given mention appears. It obtained state-of-the-art performance (micro-averaged F1-score of 83.4%) on the general-domain MSNBC dataset. A downside of this approach, however, is the significant amount of data required for training; the T-REx dataset on which CHOLAN was trained, for instance, contains more than 3 million mentions linked to 85,628 Wikipedia concepts.

To eliminate reliance on the availability of large labelled datasets, researchers have drawn inspiration from the success of Siamese networks on computer vision tasks (Minaee and Liu, 2017; Dong and Shen, 2018; Song et al., 2019; Ramachandra et al., 2020) for developing models under a *few-shot learning* setting, where only a few labelled samples are utilised during model training. Unsurprisingly, a number of papers on EL used this type of network to learn semantic representations for mentions and concepts. Neculoiu et al. (2016), for instance, trained a model based on character-level Bi-LSTMs with a Siamese architecture to learn the similarity between two given text sequences. The model was then applied to the task of job title normalisation.

Fakhraei et al. (2019) proposed another approach called NSEEN, which is also based on Siamese networks with character-level Bi-LSTM layers, but incorporated two innovations. First is the use of hard negative mining: a technique for including difficult negative training examples, so that a model can learn to distinguish them from positive examples even if they bear similarities to each other. Second is the application of an optimised *k*-nearest neighbours (KNN) algorithm called Annoy (Approximate Nearest Neighbors Oh Yeah!)[5] on the learned semantic representations, to allow for more efficient identification of best-matching concepts during inference time. Another approach, known as ELSR (Entity Linking based on Sentence Representation), which was also inspired by Siamese networks, was proposed by Jia et al. (2021). Instead of Bi-LSTM layers, their Siamese network fine-tunes a BERT model to generate sequence representations (i.e., sentence embeddings) that capture similarities between the context (i.e., the text where a mention of interest appears) and the description of a concept as provided by Freebase. When compared with other approaches based on Siamese networks underpinned by Bi-LSTM layers, ELSR demonstrated superior performance, obtaining accuracy scores of 92.09% and 84.17% on the general-domain AIDA-B and KBP2017 datasets, respectively.

In developing our own deep learning-based approaches to food EL (presented in Section 5.2), we drew inspiration from NSEEN and ELSR, investigating both Bi-LSTM-based and sentence embedding-based Siamese networks and applying Annoy to identify best-matching concepts more efficiently.

3. Research questions

The overarching aim of our work is to advance the state-of-the-art in EL in the food domain. As shown in the preceding section, previously proposed approaches to EL in the food domain employed methods that are either unsupervised (e.g., those based on string similarity and clustering) or based on supervised learning using traditional machine learning-based algorithms such as SVMs and Bayesian modelling. Meanwhile, more recent developments in deep learning have enabled researchers to achieve state-of-the-art EL performance in the general domain (e.g., on the Wikification task). As deep learning-based approaches have yet to be explored with respect to the food EL task, we seek to investigate how deep learning can be exploited for the said task.

It is, however, worth noting that due to the scarcity of datasets containing food EL annotations (as discussed in Section 2.1), it is not viable to rely on the availability of many labelled examples for training models. Even the construction of a new dataset specifically for food EL will not necessarily produce many examples of mentions linked to every concept in a knowledge base. As we will describe in Section 4, after the labelling of food mentions in 468 recipes, the majority of the concepts in our knowledge base were not linked to any mentions (and hence have no examples); the other concepts that do have linked mentions have only fewer than four examples on average.

We thus argue that few-shot learning, which allows for training a model in a supervised manner even with just a few examples, lends itself well to EL in the food domain. Instead of requiring that every concept in the target knowledge base is represented by many examples of linked mentions, few-shot learning facilitates the learning of similarity between any given mention and concept generally. Consequently, a model could learn to link a mention to a concept even if it had seen only a few (or no) examples for that concept during training.

There are also further considerations when it comes to applying a trained EL model to a real-world scenario, e.g., when retrieving external knowledge on ingredients in a recipe in real time. For a food EL solution to be truly useable at scale, it should not only output accurate predictions but also provide those predictions within as little response time as possible. Considering the above, our work is focussed on addressing the following research questions.

[5] https://github.com/spotify/Annoy

RQ1: How can state-of-the-art deep learning-based methods be exploited for EL in the food domain, and how well do such methods perform?
In the way of training deep learning-based models within a few-shot learning setting, we investigate two types of Siamese networks, one built with bidirectional LSTMs (Bi-LSTMs) and the other underpinned by pre-trained transformer models. In both approaches, semantic information is utilised by our Siamese network by means of embedding representations (static embeddings in the former and contextual embeddings in the latter). Using accuracy as an evaluation metric, we assess the performance of our Siamese networks.

RQ2: How does a deep learning-based approach to food EL compare to previously reported solutions?
To enable us to perform comparisons between our deep learning-based approach and previously reported approaches to EL in the food domain, we also develop unsupervised (string similarity-based) and traditional machine learning-based (SVM) approaches. The performance and the inference speed of these different approaches are compared by evaluating them on the same dataset, i.e., our own newly constructed food EL dataset (described in the next section).

RQ3: How can the strengths of different types of approaches be combined to build a solution to the food EL problem that is optimised for both performance and speed?
We identify the strengths of each of our developed approaches and integrate them into one hybrid model demonstrating optimal performance and speed.

To the best of our knowledge, ours is the first work to: (1) systematically compare various approaches to EL in the food domain, with respect to their performance and speed; (2) investigate few-shot learning as a paradigm for training deep learning-based food EL models, using a recipe dataset with a limited number of labelled samples (< 1000); and (3) propose a novel hybrid model that combines the strengths of approaches based on unsupervised, traditional machine learning-based and deep learning-based models, as identified through our systematic evaluation.

4. Data preparation

To help us address our research questions, we constructed a new dataset consisting of cooking recipes in which ingredient names have been manually linked to their corresponding canonical names (i.e., concepts) in a food knowledge base. Below, we describe the development of the said knowledge base and the annotation of our dataset of recipes, which we will later on refer to as the E.Care dataset.

4.1. Food knowledge base construction

While there exist comprehensive food knowledge bases such as the FoodOn Ontology (Dooley et al., 2018) and the Food and Nutrient Database for Dietary Studies (U.S. Department of Agriculture, 2019), their scope is too broad for the purposes of our E.Care project, which was aimed at estimating the carbon footprint of food within the UK context (as discussed in Section 1).

We thus constructed our own food knowledge base, the E.Care KB, building upon the Composition of Foods Integrated dataset (CoFID) (Public Health England, 2015), which contains food items relevant to the UK population. The most recent version of the CoFID dataset contains a total of 2887 food names.[6] Many of these food names, however, were considered to be duplicates of each other: they pertain to various forms of the same ingredient, e.g., 'Almonds, flaked and ground', 'Almonds, toasted', 'Almonds, weighed with shells' and 'Almonds, whole kernels'. Our use case for EL (i.e., estimating carbon footprint) does not require such level of granularity, as the scope of the E.Care project is limited to estimating the carbon footprint of food without yet taking into account any processes that a food item has undergone. For instance, we consider toasted almonds, flaked/ground almonds and whole almonds as all having the same carbon emissions, thus there is no need to distinguish between them in our knowledge base. In the example given above, for instance, only the name 'Almonds' was retained.

Some of the names in the CoFID dataset pertain to dishes rather than to individual ingredients. Examples of these include 'Curry, red kidney bean, Gujarati, homemade' and 'Cauliflower with onions and chilli pepper, homemade'. In most cases, these entries are clearly described as dishes or recipes; for example, the above two examples are respectively described as 'recipe, Bangladeshi dish' and 'recipe, thin-medium consistency' in the `Description` column of the dataset. As our knowledge base is intended to facilitate linking of ingredient names, CoFID entries pertaining to dishes were removed by filtering out food names for which the `Description` field contains the word 'recipe'. However, inspection of the remaining names revealed that not all dish names were removed, as some of them (e.g., 'Doner kebab in pitta bread with salad') were not described as recipes nor dishes in the dataset. To address this issue, we manually checked all remaining food names, ensuring that only individual ingredient names were kept.

The steps described above resulted in an initial version of the E.Care KB containing 532 canonical food names. Each of these canonical names was manually matched to the closest term in the FoodOn Ontology, a comprehensive, standard vocabulary for describing food, which is widely used by the food data science community. This step was carried out for two purposes: (1) to assign the most suitable FoodOn identifier to every canonical name, in order to foster interoperability of the E.Care KB with other, future applications; and (2) to harvest any alternative names in FoodOn that are associated with every canonical name, in order to enrich the E.Care KB with synonyms. Overall, 1027 synonyms were added to the knowledge base. It is worth noting that this initial version of the knowledge base has since been expanded based on food names that were identified as missing during manual annotation of recipes (described below).

[6] https://www.gov.uk/government/publications/composition-of-foods-integrated-dataset-cofid

4.2. Semi-automatic annotation of recipes

As a first step towards the creation of a corpus of recipes, we designed a survey intended to collect a list of online cooking recipes from UK participants. Together with a Participant Information Sheet and an offer of compensation (i.e., a voucher of choice), the survey was disseminated through our professional networks as well as social media. This was done without disclosing that the survey was part of a project aimed at estimating the carbon footprint of food, in order to avoid influencing or biasing responses from participants. A total of 132 participants – representative of different age, gender, ethnicity and income groups – took part in the survey, each one specifying the URLs for their five favourite recipes, i.e., dishes that they usually cook. After removal of recipes that are not written in English and removal of duplicates across all participants, a final list of 587 recipes was obtained. In order to annotate the ingredients contained in these recipes by linking them to their corresponding canonical names in the E.Care KB, the steps described below were taken.

4.2.1. Parsing of recipe web pages

A third-party web service offered by Spoonacular[7] was utilised to automatically extract the recipe title and list of ingredients from a given recipe web page (specified by its URL). The web service returned its output in JSON format. Unfortunately, not all 587 recipe web pages could be parsed by the web service; acceptable output was produced for only 468 URLs. In many of the failed cases, the Spoonacular web service was unable to locate the section of the web page that contains the list of ingredients, and hence did not return any information that we could include in our dataset.

4.2.2. Named entity recognition

A prerequisite step for EL is the recognition of food names (i.e., mentions) within specified ingredients. For example, in the ingredient '100 g bacon lardons', the food name 'bacon lardons' needs to be extracted before it can be linked to 'Bacon' in the food knowledge base. To this end, we developed our own NER model by fine-tuning a transformer-based language model (Devlin et al., 2018) – in particular the `bert-large-cased` implementation[8] – for the NER task based on the FoodBase recipe corpus (Popovski et al., 2019c) which contains 1000 recipes with manually annotated food names. We divided the corpus into training, validation and test subsets following a 70–10%–20% split. After utilising the training and validation subsets for fine-tuning, the resulting NER model obtained the following performance on the food names in the test subset: 97.24% for precision, 97.70% for recall and 97.47% for F1-score.

The newly developed food NER model was applied on the list of ingredients extracted from each of the 468 recipes that were automatically parsed as part of the process described in Section 4.2.1. Specifically, the NER model returned a recognised food mention for every given line of text that corresponds to an ingredient item in a recipe's ingredient list. In cases where a specified ingredient item also includes alternative ingredients (e.g.,'1 1/2 cups green lentils or brown lentils'), the NER model returned more than one recognised food mention (e.g., 'green lentils' and 'brown lentils').

For every recipe, a comma-separated values (CSV) file was generated, whereby the original ingredients in the recipe were listed in one column and the corresponding food names automatically recognised by the model were presented in another column. An annotator was then employed to manually correct any erroneous predictions by the NER model, either by filling in any mentions that were missed or by changing/adjusting the span of the food name that was recognised (e.g., in the ingredient item '5 cloves of garlic', the automatically recognised name 'cloves' was manually corrected to 'garlic'). Out of a total of 6391 ingredients across all 468 recipes, 6282 of the recognised food mentions were identified as true positives (correctly recognised names), 87 were false positives (token sequences that were incorrectly recognised as food names) and 22 were false negatives (food names that were missed).

4.2.3. Manual entity linking

The now validated food mentions formed the basis of the E.Care dataset of recipes, where each food mention is linked (i.e., normalised) to the best-matching canonical name in the E.Care KB. To this end, two annotators were employed to undertake the manual linking task, which was divided into two phases. First, the annotators were asked to independently find within the knowledge base, the best-matching canonical name for every food mention in each of the 468 recipes. For this, the annotators were given access to both the canonical names and the corresponding synonyms in the E.Care KB, and were allowed to look up (e.g., in Wikipedia) any food mentions that they might be unfamiliar with. Upon completion of this task, we calculated the agreement between the two annotators based on Cohen's Kappa (Cohen, 1960) and obtained 93%, which is considered to be almost perfect agreement according to Landis and Koch (1977).

The second phase was concerned with harmonising the EL labels from the two annotators, who were asked to review the food mentions on which they did not agree during the first phase. This time, they were allowed to have discussions with each other in order to reach a consensus. Thus, full (100%) agreement on the manual EL task was obtained in this phase.

[7] Endpoint located at https://spoonacular.com/food-api/docs#Extract-Recipe-from-Website

[8] Model available at https://huggingface.co/bert-large-cased. This pre-trained language model was fine-tuned for NER (token classification) using the Python scripts at https://github.com/huggingface/transformers/tree/main/examples/pytorch/token-classification.

Table 1
Characteristics of the E.Care resources.

E.Care KB	Number of concepts	664
	Number of synonyms	1130
E.Care Dataset	Number of recipes	468
	Number of unique food mentions	935
	Number of ingredients	6441
	Number of words	30,076
	Average number of ingredients per recipe	13.76
	Average number of words per recipe	64.26
	Average number of words per ingredient	4.67

4.3. The E.Care knowledge base and the E.Care dataset

After the addition of the food names that were identified as missing during manual EL (Section 4.2.3), the final version of the E.Care KB contains 664 canonical names (i.e., concepts), which are all linked to their respective closest matching terms in the FoodOn Ontology, and 1130 synonyms. The concepts are organised as a flat list, i.e., without capturing any hierarchical relationships between them. The data in our knowledge base is programmatically accessible via application programming interface (API) endpoints that were implemented using the Django web framework.[9]

The E.Care dataset of 468 recipes, meanwhile, consists of 935 unique food mentions, each of which is linked to its canonical name in the E.Care KB. It is worth noting that not every canonical name in the knowledge base was linked to any food mention in the dataset. Out of the 664 canonical names (concepts) in the E.Care KB, 452 concepts were not linked to any mentions in the E.Care dataset, i.e., they were not referred to in any of the recipes. Of the other 212 concepts, the average number of linked mentions is 3.74. Table 1 presents a summary of the characteristics of the E.Care KB and the E.Care dataset. In Appendix A, we provide an example list of ingredients together with the KB concepts that the contained food mentions are linked to.

5. Methods

In this section, we describe in detail how our various approaches to EL were developed. We start by presenting details of the pre-processing techniques applied to the E.Care dataset (described in the previous section). We then describe our two deep learning-based approaches. This is followed by a discussion of the traditional machine learning-based and unsupervised approaches that we implemented, to allow for comparison between the different approaches to EL. Fig. 1 provides a visual summary of our overall methodology; it is worth noting that the comparison, evaluation and combination of the different approaches will be discussed in Section 6.

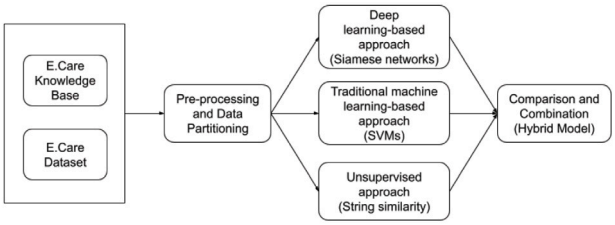


Fig. 1. A visual depiction of our EL methods.

5.1. Pre-processing and partitioning of the data

Our pre-processing stage involved multiple techniques that were carefully selected based on experimentation. Specifically, it involved applying the following steps on each of the 935 food mentions in our dataset: (1) case-folding; (2) stop word removal using Gensim's stop word list and library;[10] (3) removal of non-alphanumeric characters, punctuation, numbers, multiple consecutive whitespace characters and short tokens (with length less than or equal to two characters); and (4) lemmatisation of nouns using NLTK's WordNet lemmatiser.[11]

To generate examples for training and evaluating our approaches, each of the 935 mentions was paired up with the canonical name in the knowledge base to which it is linked to. Following a 50–50% split, half of the pairs were designated as training examples; the other half were set aside for evaluation. The set of training examples was then expanded by pairing up each of the mentions with every synonym of the linked canonical name. This resulted in a total of 1821 pairs which are considered as positive examples,

[9] https://www.djangoproject.com/
[10] https://tedboy.github.io/nlps/_modules/gensim/parsing/preprocessing.html
[11] https://www.nltk.org/_modules/nltk/stem/wordnet.html

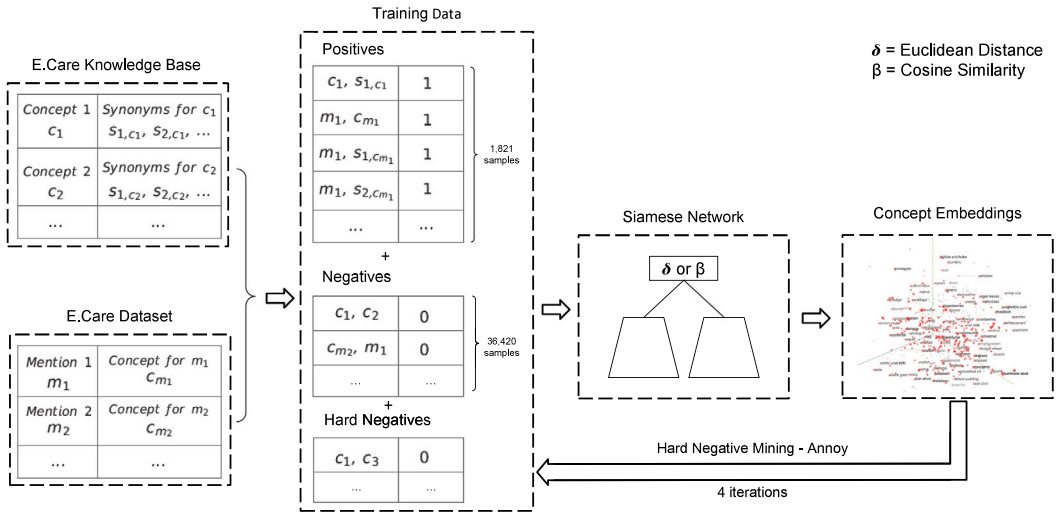


Fig. 2. Framework for training Siamese networks. Key: m_i: mention i from the E.Care dataset; c_{m_i}: concept corresponding to mention i; c_i: concept i from the E.Care KB; s_i, c_j: synonym i corresponding to concept j.
Source: Image reproduced from Fakhraei et al. (2019).

i.e., mentions correctly linked to their corresponding concepts. It is, however, worth noting that as discussed in Sections 5.2.1 and 5.3.1, the training set for our traditional machine learning-based and deep learning-based approaches was further enriched by the inclusion of negative examples, i.e., mentions paired up with incorrect concepts and labelled accordingly.

5.2. Deep learning-based approach

Inspired by the recent success of deep learning-based techniques (see Section 2.4.3), we developed and compared two models based on Siamese networks to detect similarity between a pair consisting of a mention and a concept in the E.Care KB. The first model is underpinned by Bi-LSTM layers, while the second one is based on pre-trained sentence transformers.

A Siamese network consists of two branches with tied weights. That is, two copies of the same network are merged based on a similarity (or distance) function (Neculoiu et al., 2016). The input token sequences, in this case the mention and the concept, are represented as embeddings in the same space. The end-goal of this type of networks is to learn embedding representations for the input sequences, which are then used to detect the similarity between them.

5.2.1. Preparing to train siamese networks

In order to enable a Siamese network to learn similarity between a given mention and a concept, examples of pairs where the mention and the concept are similar (labelled as 1), and pairs where they are dissimilar (labelled as 0), are necessary. To this end, we took the 1821 positive examples designated for model training, which we described in Section 5.1. With regard to generating negative examples, we took inspiration from the work of Fakhraei et al. (2019) which demonstrated remarkable performance on the Bio-ID dataset (Arighi et al., 2017). Specifically, every mention in the set of positive examples, was paired up with each of 20 randomly selected concepts in the knowledge base. This resulted in the creation of an initial training set of 38,241 examples, of which 1821 are positive and 36,420 are negative. However, we also employed a technique known as *hard negative mining* in order to identify negative examples that are most informative for the model. These examples are the ones that are the closest to the decision boundary in the embedding space, and thus the ones that the model would most likely classify incorrectly. Previous work demonstrated that this technique improves the performance of Siamese networks on similarity detection tasks (Liang & Shen, 2019). Fig. 2 provides a summary of the overall framework that we adopted in training each of our Siamese networks.

5.2.2. Siamese Bi-LSTM network

In our first Siamese network, depicted in Fig. 3, each of the two branches consists of a Bi-LSTM layer and a dense feedforward layer. The input to the Bi-LSTM layer are two vector representations: one for the mention and another for the concept. These are static embeddings that were obtained from a fastText model (Bojanowski et al., 2017) that we had pre-trained on the Recipe1M+ dataset (Marin et al., 2019). The similarity between the two output vectors v_i and v_j (of the two dense feedforward layers) is then computed based on Euclidean distance following the work of Shih et al. (2017).

The output of the entire network is a value that is greater than or equal to 0, representing the similarity between the mention and the concept that were given as input. Contrastive loss is then employed to update the model weights, as shown in the following equation:

$$\mathcal{L} = \frac{1}{2} y \cdot \delta(v_i, v_j)^2 + \frac{1}{2} \cdot (1 - y) \cdot max(0, 1 - \delta(v_i, v_j))^2$$

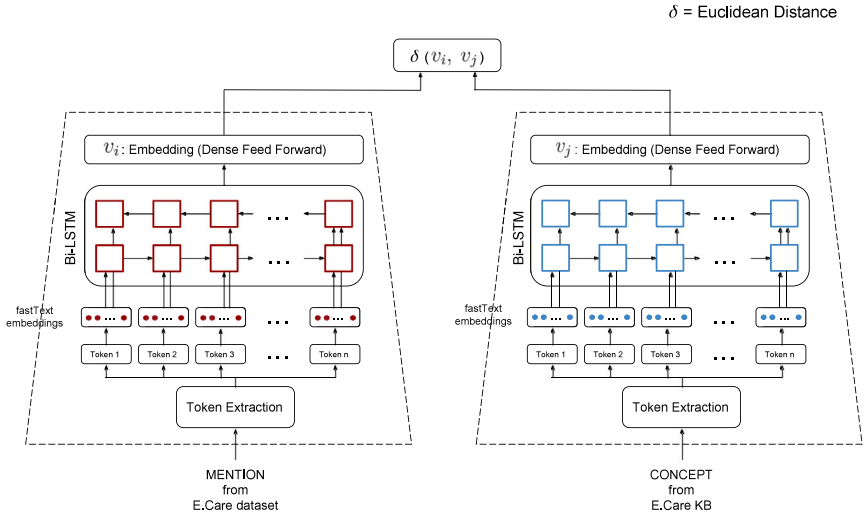


Fig. 3. The Siamese Bi-LSTM architecture we adopted.
Source: Image reproduced from Fakhraei et al. (2019).

where $y = 0$ if the example is negative, otherwise $y = 1$, and $\delta(v_i, v_j)$ represents the Euclidean distance between the vector representations of the given mention and concept. The goal of this loss function is to bring closer together in high-dimensional space examples that belong to the same class (i.e., positive examples with label $y = 1$), while pushing examples from the other class (negative examples with label $y = 0$) farther away from the former by a margin m, which was set to 1 in our work.

As mentioned in Section 5.2.1, hard negative mining was employed in order to select negative examples that are most informative for the model. First, we trained our model on the initial training dataset; then, we utilised our static embeddings to represent all of the concepts and synonyms in the E.Care KB onto the trained embedding space. We then employed an optimised KNN algorithm, i.e., Annoy, to find the concepts (or synonyms) in the knowledge base that are closest to a given mention. Finally, we added those mention-concept pairs as negatives (with the label set to 0) to our training set and retrained our model. We repeated this process four times to refine the model with multiple sets of hard negatives.

5.2.3. Siamese network with sentence transformers

Our second Siamese network is similar to the first one in that its goal is also to predict a similarity score for a mention-concept pair. Thus, the same dataset used in training the Siamese Bi-LSTM network was utilised in training the second network. However, instead of using static embeddings to represent each of the mention and the concept (as in the first network), we employed embeddings computed by pre-trained sentence transformer models (Reimers & Gurevych, 2019) that have demonstrated state-of-the-art performance in semantic textual similarity (STS) tasks (Ha et al., 2021). Additionally, the Bi-LSTM layer was replaced with a pooling layer. We refer the reader to Fig. 4 for a diagram depicting the Siamese network based on sentence transformers.

Three pre-trained sentence transformer models were selected based on the reported performance on the task they were originally trained for, and their encoding speed (i.e., the time it takes the models to compute the embedding vector). Two of our chosen models were trained for paraphrase mining: `paraphrase-MiniLM-L3-v2`[12] and `paraphrase-MiniLM-L6-v2`.[13] They have a speed of 19,000 and 14,200 encodings per second, respectively, and have obtained an average performance of 62.29% and 64.82%, respectively, on 14 datasets. Additionally, since Conneau et al. (2017) demonstrated that training sentence embedding models on natural language inference (NLI) data leads to improved results, we also made use of the `nli-distilroberta-base-v2` sentence transformer model[14] that was trained for NLI tasks; it has a speed of 4000 encodings per second and obtained an average performance of 84.38% on the STS benchmark (STSb) dataset.

The training process is analogous to that of the Siamese Bi-LSTM network: first, for each of the mention and the concept in every training pair, an embedding representation was obtained using a pre-trained model's encoder. The same pre-trained model was then fine-tuned for the similarity learning task. Hard negative mining based on Annoy was then employed to generate difficult negative examples. These examples were then added to the training set for the next training iteration. This process was repeated four times to refine the model.

[12] https://huggingface.co/sentence-transformers/paraphrase-MiniLM-L3-v2
[13] https://huggingface.co/sentence-transformers/paraphrase-MiniLM-L6-v2
[14] https://huggingface.co/sentence-transformers/nli-distilroberta-base-v2

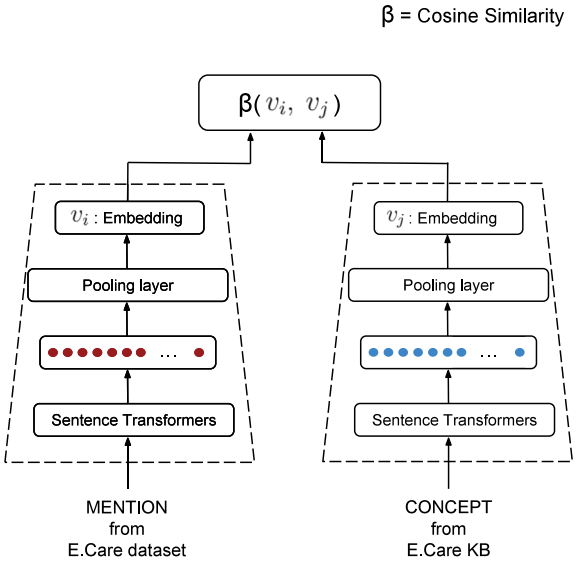


Fig. 4. Architecture of the Siamese network underpinned by sentence transformers.

5.3. Traditional machine learning-based approach

As discussed in Section 2.4.2, traditional machine learning algorithms formed the basis of previously proposed approaches to EL in the food domain. To allow for comparison with our deep learning-based approaches, we also implemented an EL solution that is underpinned by SVMs. Specifically, an SVM model was trained as a binary classifier to determine whether a mention (from recipe text) and a concept (from our knowledge base) are similar or not.

5.3.1. Preparing to train SVMs

As with our deep learning-based approaches, positive examples ('similar' pairs) and negative examples ('dissimilar' pairs) are required for training an SVM model. The process for generating positive examples is similar to that discussed in Section 5.2.1. However, instead of relying on embeddings generated by neural models (as we did in preparing data for training Siamese networks), we made use of string similarity indices to identify hard negative examples. Specifically, we employed the widely-used Jaro distance to measure string similarity between a given mention and every concept (or synonym) in our knowledge base. To ensure that challenging cases are included as negative examples, the 150 highest-scoring mention-concept pairs were selected. For example, for the mention 'tamari', this method will lead to the identification of 'tamarind' as a negative example, which is desirable, since 'tamari' and 'tamarind' are lexically similar yet they are unrelated and share no semantic similarity. In the end, our training set included a total of 100,719 examples, of which 1821 are positive and 98,898 are negative.

5.3.2. Features for training SVMs

SVMs require examples to be represented as hand-crafted features. Prioritising features that can be generated at speed, we chose the following: (1) bag-of-words (BoW) representation of each of the mention and the concept in a pair; (2) lexical similarity between the mention and the concept, computed based on overlapping bigrams; and (3) semantic similarity which was computed based on the cosine similarity between the embedding representations of the mention and the concept, obtained from a fastText model that we pre-trained on the Recipe1M+ dataset.

5.4. Unsupervised approach

In addition to deep learning-based and traditional machine learning-based approaches to EL, we also implemented unsupervised approaches. Specifically, a given mention – pre-processed as described in Section 5.1 – is paired up with every concept (and synonym) in the E.Care KB. The similarity between the mention and the concept is calculated based on a string matching algorithm; the concept for which the highest score was obtained is then taken as the best match. Additionally, if a mention consists of only two tokens, they are swapped with each other to form a reordered version of the mention. This step is helpful in cases where the food mention is a noun phrase consisting of a head noun and its modifier (e.g., 'wheaty chorizo', 'chopped capsicum') but the corresponding concept

Table 2
Evaluation results for deep learning-based models. Best results are shown in bold.

Model for Siamese network	Embeddings	Token removal?	Accuracy[a] (%)	Standard deviation	Time (sec/mention)
Bi-LSTM	fastText	Yes	82.55	0.89	0.039
		No	76.03	1.28	0.032
Sentence transformers	nli-distilroberta-base-v2	Yes	85.81	1.02	0.016
		No	79.16	1.08	0.01
	paraphrase-MiniLM-L3-v2	Yes	**86.99**	0.47	0.007
		No	83.20	1.52	**0.004**
	paraphrase-MiniLM-L6-v2	Yes	85.86	0.37	0.01
		No	81.75	1.35	0.007

[a]Represents mean accuracy computed by training and evaluating each model five times.

(or its synonym) in the E.Care KB consists only of the head noun (e.g., 'chorizo', 'capsicum'); reordering the two tokens can lead to a similarity score that is higher than if the original order of tokens was used. The matching process described above is then also applied to the reordered version of the pre-processed mention. The match that obtained the highest score is finally taken as the linked concept. We refer the reader to Appendix B for a diagram illustrating this process.

Eight different string matching algorithms were investigated: Jaro, Jaro–Winkler, Jaccard, Trigrams Cosine, Q-grams, Levenshtein, LCS and Novel Bigram (Kaur, 2015).

6. Evaluation and combination of models

In this section, we present the experimental setup that was employed in this research, as well as the results of evaluating each of the approaches described in the previous section on the partition of our dataset that was set aside as our test set. Furthermore, we describe and evaluate a number of hybrid models that were built by combining the strengths of the deep learning-based, traditional machine learning-based and unsupervised approaches.

6.1. Experimental setup

Following the convention used in previous EL work, we report the performance obtained by each of the approaches in terms of accuracy: the number of correctly linked mentions over the total number of mentions in the test set. To mitigate the impact of random parameter initialisation and the stochastic nature of optimisation algorithms on model performance, each of our models was trained five times. This allows us to report averaged accuracy and standard deviation, which provide a more reliable assessment of model performance. Importantly, in order to assess the scalability of each of the approaches in terms of inference speed, we also report the average time it takes to link a given mention.

Before evaluation, two steps were applied to the test set. First, to avoid biasing the performance results towards 'easy' examples, we removed mention-concept pairs where similarity is obvious or straightforward; these included cases where the mention and the concept are exactly the same, e.g., *(pickle, pickle)*, or where they differ only in terms of one being the singular or plural form of the other, e.g., *(tomatoes, tomato)*. This process resulted in a total of 406 unique mentions in the test set.

Second, we introduced a token removal step whereby we eliminated rare words (e.g., 'higher welfare' in 'higher welfare bacon') or modifiers that are unlikely to appear in the canonical names in the knowledge base. As explained in Section 4.1, the E.Care KB does not distinguish between different forms or variations of the same food item. Hence, modifiers such as adjectives (e.g., 'fresh' in 'fresh parsley') that tend to appear in a recipe's ingredients list, are unlikely to appear in the canonical names. To eliminate modifiers, any non-nouns in a given mention (identified using NLTK's POS-tagger[15]) were removed. Additionally, token-level similarity checking was performed, in which each noun token (identified using the same POS-tagger) was compared with each token in every concept or synonym in the knowledge base. A token was considered rare and thus removed if the highest similarity obtained using Jaro distance was less than a threshold that was set to 0.9 based on experimentation. If this process led to all tokens in a mention being removed, the original mention was used.

It is worth noting that in reporting the average time it takes to link a given mention, we are including the time required to pre-process the mention, as well as the time for the token removal step just described. To ensure consistency and comparability of our results, all our models were trained and evaluated on a server equipped with an NVIDIA Tesla T4 GPU with 16 GB of memory.

[15] https://www.nltk.org/book/ch05.html

Table 3
Evaluation results for the SVM model. Best results are shown in bold.

Model	Token removal?	Accuracy (%)	Time (sec/mention)
SVM	Yes	86.45	9.86
with exhaustive comparison	No	**87.19**	29.06
SVM	Yes	**87.19**	**0.66**
with filtered concepts	No	84.23	2.38

6.2. Deep learning-based approach

As described in Section 5.2.2, our two deep learning-based methods are underpinned by Siamese networks, one based on a Bi-LSTM network fed with static embeddings and the other based on sentence transformers. Our deep learning-based models were trained for 20 epochs (followed by 4 iterations of hard negative mining) using the Adam optimiser, a batch size of 16, with a learning rate of `1e-3` for the Bi-LSTM model and `2e-5` for the sentence transformer models. In each of the methods, Annoy was used to filter out candidates in order to improve performance in terms of speed. Here, we selected only the five candidate concepts that are most similar to a given mention. Selecting more candidates ($n > 5$) was also explored; however, our experimentation showed that increasing the number of candidates does not lead to improved accuracy. As shown in the results presented in Table 2, our token removal step (with similarity threshold set to 0.9) leads to a significant improvement in accuracy (up to 6.65 percentage points), when it comes to the performance of the `nli-distilroberta-base-v2` model or the Siamese Bi-LSTM network fed with static embeddings. Furthermore, for each model, a low standard deviation was calculated over different runs, indicating a consistent and stable performance with minimal variance. Notably, the paraphrase models, which obtained the highest accuracy among all the models (86.99% and 85.86%), also exhibited the lowest standard deviation (0.47 and 0.37).

As for the Siamese network fed with pre-trained sentence transformer embeddings, it can be observed in Table 2 that using `paraphrase-MiniLM-L3-v2` to encode a mention-candidate pair leads to slightly improved accuracy (86.99%) compared to using `paraphrase-MiniLM-L6-v2` (85.86%) or `nli-distilroberta-base-v2` (85.81%). Moreover, it is evident that employing `paraphrase-MiniLM-L3-v2` for encoding leads to faster linking (0.007 sec/mention), i.e., more than twice as fast as `nli-distilroberta-base-v2` (0.016 sec/mention).

6.3. Traditional machine learning-based approach

As described in Section 5.3, an SVM model was developed to perform binary classification. Our model was implemented using the `scikit-learn` library.[16] A number of parameters were configured such as: type of kernel, i.e., linear, polynomial or radial basis function (RBF); a regularisation parameter C that trades off correct classification of training samples against maximisation of the decision function's margin; and gamma, a parameter that defines how far the influence of a single sample reaches. Our experiments showed that the best performing SVM model is obtained by training it using an RBF kernel (which is capable of handling non-linear data) and by setting the regularisation parameter C to 1 and the gamma value to 0.7.

A bottleneck in casting the EL problem as a binary classification task is the need for exhaustive comparison, i.e., pairing up a given mention with each of the concepts in the knowledge base prior to classification. We thus sought to eliminate concepts that are unlikely to be similar to a given mention. To this end, we employed our own fastText model (pre-trained on the Recipe 1M+ dataset) to obtain an embedding representation for each concept in the E.Care KB, as well as for a given mention. For a given mention-concept pair, the cosine similarity between the embedding representations of the mention and the concept was then computed; all concepts for which a similarity score of less than 0.55 was obtained were eliminated from being candidates. As shown in Table 3, the SVM model that is provided with filtered concepts achieve the same accuracy (87.19%) as the SVM model with exhaustive comparison, but is approximately 44 times faster, with a linking time of 0.66 sec/mention compared to 29.06 sec/mention.

6.4. Unsupervised approach

As described in Section 5.4, for our unsupervised approach, we implemented and compared eight different string matching algorithms. These string similarity algorithms were applied to two versions of a given mention: the original one and one resulting from our token-removal step; for the latter, a similarity threshold of 0.9 was chosen based on experimentation. The mention is then linked to the concept with the higher similarity score, based on comparing the two scores obtained by using the two different mention versions as input. From Table 4, one can observe that the best performing string matching method, in terms of accuracy, is that based on cosine similarity (87.43%). It is worth noting that in linking mentions based on our unsupervised approach, synonyms of concepts in the E.Care KB were not taken into account as our experiments showed that including them as candidates harms accuracy (see Appendix C for the results of these experiments).

[16] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

Table 4
Evaluation results for string matching methods using a similarity threshold of 0.9 and the Jaro index for token removal. Best results are shown in bold.

Algorithm	Accuracy (%)	Time (sec/mention)
LCS	66.01	0.007
Jaccard	68.22	0.006
EDIT	71.42	**0.005**
Jaro–Winkler	82.51	0.02
Jaro	82.75	0.02
Novelty Bigram	85.71	0.18
Q-gram	86.69	0.03
Cosine	**87.43**	0.02

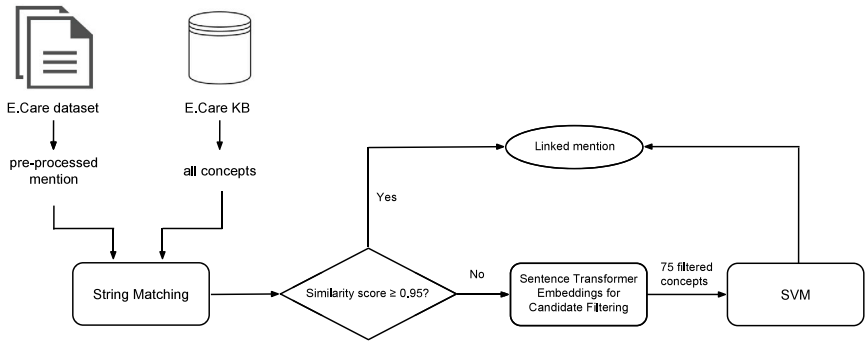


Fig. 5. Pipeline of our hybrid model.

6.5. Hybrid model

Based on the results of our evaluation (as reported in the previous section), we summarise below the strengths and weaknesses of each of the approaches we have developed with respect to their performance and speed (i.e., the average time it takes a model to link a mention to a concept):

- The unsupervised approach based on cosine similarity obtains the highest performance (87.43%) in terms of accuracy. It is able to link mentions at speed, with an average linking time of 0.02 seconds per mention. However, by definition, it takes into consideration only lexical similarity.
- The traditional machine learning-based approach based on an SVM model trails behind the cosine similarity-based unsupervised approach by a small margin in terms of accuracy (87.19% vs 87.43%). The features that were chosen to represent every mention-concept pair were designed to capture both lexical and semantic similarity. Its average linking time, however, can be impractical (i.e., 29.06 seconds per mention without candidate filtering).
- The deep learning-based approach underpinned by sentence transformer embeddings trails behind the SVM model in terms of accuracy (86.99% vs 87.19%), but is fastest at linking mentions with an average linking time of as little as 0.004 seconds per mention. This can be attributed to the incorporation of Annoy as a technique for filtering out unlikely candidates and choosing as few as only five for every given mention, which significantly lessens the number of comparisons required.

Our proposed solution is a hybrid model (depicted in Fig. 5) that leverages the strengths of the previously presented approaches while addressing their respective weaknesses. It combines the best performing unsupervised approach, which achieves high accuracy but considers lexical similarity only, with an SVM model that captures both lexical and semantic similarity but requires longer linking time. To reduce linking time, we also integrated the best performing deep learning-based approach, which utilises sentence transformer embeddings and employs filtering techniques to decrease the number of comparisons required, which we set to 75. By combining these approaches, our unified hybrid model offers improved overall performance.

Our hybrid model works as follows: first, the unsupervised model is used to measure the similarity between a given mention and a concept in the knowledge base. If the similarity score is greater than or equal to 0.95, the mention is linked to the concept; otherwise, the mention is processed by the SVM model. To eliminate the exhaustive comparison required by our SVM model, we made use of candidate filtering whereby Annoy was used to select 20 candidates that are most similar to a given mention. Here, to determine the type of embedding representation that works best for filtering candidates, we compared the embeddings learned by our Siamese Bi-LSTM model, embeddings obtained from the `paraphrase-MiniLM-L3-v2` sentence transformer model, as well as static embeddings from our fastText model.

Table 5
Evaluation results for hybrid models.

Embeddings for candidate filtering	Accuracy (%)	Time (sec/mention)
fastText	89.16	0.79
Siamese Bi-LSTM	88.17	0.55
paraphrase-MiniLM-L3-v2	**89.40**	**0.24**

As presented in Table 5, the results of our experiments show that the best performing hybrid model is the one that employs `paraphrase-MiniLM-L3-v2` sentence transformer embeddings for candidate entity generation. It achieves an accuracy of 89.40%, with an average linking time of 0.24 seconds per mention.

7. Discussion of results and implications

In this section, we analyse the performance of our various models for food EL and provide a discussion of the trade-offs between their accuracy and linking time, as well as the amount of labelled data that they require. This is followed by some error analysis and an overview of the implications of our results.

7.1. Comparative analysis: Accuracy, speed and required training data

First, our unsupervised string matching algorithm which uses cosine similarity for linking and the Jaro index for token removal, achieves competitive performance in terms of accuracy and linking time (i.e., the number of seconds required to link each mention to a concept) when compared to our other models. This demonstrates that string matching algorithms can obtain satisfactory results on the food domain. This unsupervised model is perhaps the most advantageous in terms of the required amount of annotated data and training time, as it does not require any supervised training. However, the main drawback of this model is its reliance on lexical similarity only, i.e., solely on similar patterns found within the strings. This can lead to errors in matching, especially in cases where a concept is referred to using completely different strings (e.g., 'aubergine' and 'eggplant') or when the data is noisy.

Our traditional machine learning-based approach based on an SVM model achieves similar performance in terms of accuracy, but unlike the unsupervised model, it makes use of both lexical and semantic features, in which the strength of this model lies. The inclusion of semantic features such as pre-trained word embeddings, enables the SVM model to better capture the meaning of a mention, resulting in improved robustness to variations in strings. For instance, the fastText word embeddings that were used as features by our SVM model capture semantic similarities even between lexically different strings such as 'aubergine' and 'eggplant', as they would have learned (during pre-training on millions of recipes) that these two mentions tend to be used in similar contexts. Moreover, these semantic features provide a more generalisable representation of the mentions and concepts, allowing the model to perform well on new and unseen mentions. As fastText learns embeddings at the level of subwords (i.e., character *n*-grams), it can provide informative embedding representations even for mentions and concepts which were not seen during pre-training. Another advantage of this model is its capability to produce reasonable results even with only a small set of annotated data. However, its disadvantage becomes evident when applied to EL with a knowledge base with a large number of concepts, as the number of hard negatives required for model training grows quadratically. Linking time also increases proportionally, although this can be alleviated by candidate filtering. SVMs require the engineering and/or selection of features by hand. In our case, the SVM model was fed with the following features representing a given mention-concept pair: BoW representation of each of the mention and the concept, lexical similarity based on shared bigrams, and semantic similarity based on the cosine similarity between the fastText embedding representation of each of the mention and the concept. As satisfactory performance was obtained by our SVM model when trained on our chosen features, we decided to not explore any other features. In principle, however, one could invest more resources in designing other features to incorporate into the training of the SVM model, which tends to be a time-consuming task that, at times, requires domain expertise.

Our deep learning-based approach underpinned by Siamese networks obtains satisfactory results in terms of accuracy and the best results in terms of inference time, with the best performing model (which takes 0.007 seconds to link each mention) being almost three times faster than the best string matching model (the cosine-based one that requires 0.02 seconds per mention), and about 94 times faster than the SVM model with the best inference time (requiring 0.66 seconds per mention). The scalability of our deep learning-based models (in terms of inference time) can be attributed to the incorporation of the Annoy algorithm for identifying candidates. Similarly to the SVM model, our deep learning-based models require annotated data for model training; however, taking a few-shot learning approach meant that the models are able to learn the EL task even with only a few training samples per concept. Unlike SVMs which require feature engineering, the deep learning-based models make use of representations such as fastText embeddings and sentence transformer embeddings that were automatically learned. However, although deep learning models are considered state-of-the-art in NLP (as discussed in Section 2.4.3), in our case, they did not provide the best results for food EL. The relatively low performance of our first Siamese network could be attributed to the fact that we used only one Bi-LSTM layer; incorporating more Bi-LSTM layers could have possibly enabled the network to learn the similarity between a given mention

and concept better. Meanwhile, with regard to our second Siamese network, the sub-optimal performance is possibly owing to the sentence transformer embeddings (which were used as feature representation) having been trained on datasets with full sentences that were drawn from the general domain (i.e., from out-of-domain data).

Our proposed hybrid model, which unifies our best unsupervised string matching-based approach with our SVM model as well as the sentence transformer-based model for candidate filtering (i.e., the one based on `paraphrase-MiniLm-L3-v2` embeddings), obtains the best overall accuracy (89.40%) while maintaining reasonable linking time (0.24 seconds per mention). One drawback of the hybrid model lies in the fact that it requires exhaustive comparison (i.e., string similarity measurement between a mention and every possible concept in the knowledge base), although in our case, the EL process still produces its output within reasonable time and thus remains practicable. For instance, in a use case where ingredients in an online recipe are being linked to a knowledge base, even a recipe with 15 ingredients (which is slightly more than the average number of ingredients for each recipe in the E.Care dataset) will require only 3.6 seconds, which is still considered to be acceptable.[17] Unlike the purely unsupervised, string matching-based approach, our hybrid model ensures that semantic similarity is taken into consideration by both the SVM model and the candidate filtering method; this makes our solution more robust to new, unseen examples where mentions and their corresponding linked concepts might not necessarily bear any lexical similarity. On the basis of this, we posit that the slightly longer linking time required by the hybrid model (0.24 seconds per mention) compared to that of our best string matching-based method (0.02 seconds per mention) is a reasonable trade-off for the improvement in accuracy (almost 2 percentage points) obtained by the hybrid model over the purely unsupervised approach.

7.2. Error analysis

We manually analysed our best performing hybrid model's errors on our test dataset and identified five broad categories of challenging cases: (1) name variations, (2) accidental lexical similarity, (3) ambiguity, (4) insufficient context — where the model made reasonable mistakes due to the lack of context, and (5) miscellaneous — where the error does not fit into any of the four categories. In Table 6, we present the results of this analysis, where it can be observed that nearly half of the errors (20 out of the 43 examined mentions) can be attributed to ambiguity. It arises when the model links a mention to a general term instead of a more specific one (e.g., the mention 'fat soured cream' linked to 'cream' instead of 'double cream'), or when it selects the wrong higher-level concept, as illustrated in the example provided in Table 6. Furthermore, the analysis indicates that a significant portion of mistakes (18.6%) falls under the miscellaneous category, implying the presence of additional factors influencing the model's performance such as partial overlaps (e.g., character *n*-grams) between the mention and concept. Moreover, name variations contribute to 14.6% of the errors, highlighting the model's difficulty in recognising alternative names for specific concepts, despite its generally good performance. In terms of the model's ability to link previously unseen mentions, unsurprisingly, it demonstrates a high degree of success in cases where there is significant lexical similarity between the mention and the corresponding concept. Nevertheless, it is also able to link previously unseen mentions that lack such similarity. For instance, the model can link mentions such as 'Ossobuco' to the concept 'Oxtail'. This success can be attributed to the use of both the deep-learning based model for selecting high-quality candidates, and the SVM model for incorporating semantic features. The insights obtained from the analysis in Table 6 are valuable for enhancing the performance of the hybrid model, and serve as the foundation for our proposed future work, which is discussed in Section 8.

7.3. Theoretical and practical implications

From a theoretical perspective, our work in developing EL models and resources for the food domain advances the state-of-the-art in food data science by proposing supervised models based on traditional machine learning and deep learning, as well as a hybrid model that combines three different approaches. This is in contrast to majority of the related work in EL for the food domain, which has mainly focused on unsupervised methods relying on string matching algorithms. Our efforts not only provide a framework for researchers to investigate EL approaches for specialised domains like food, but also open up new challenges for researchers to address, e.g., how performance can be further improved by integrating semantic relationships that are codified in food-related ontologies into EL methods.

While the NLP task of EL has been explored well in the general domain (as in the case of Wikification), developing EL models intended for specialised domains can be a daunting task. First, a relevant knowledge base needs to be identified, if not constructed. A dataset of documents in which mentions have been linked to their canonical names in the relevant knowledge base is also required, for the purposes of model development (in the case of supervised methods) and evaluation. The development of our models demonstrates that traditional machine learning-based and deep learning-based methods can be applied successfully to EL in the food domain. This highlights the potential for further advancements in EL by exploring alternative approaches and combinations of approaches, particularly for specialised domains that have been relatively under-explored.

Furthermore, the wide range of types of approaches to EL can pose a challenge to researchers who wish to identify the approach that is most suitable to a domain of interest. Our work provides a blueprint for completing the steps necessary to build an EL solution, while also highlighting the strengths and weaknesses of each potential approach. Moreover, we demonstrate how the strengths of

[17] According to https://www.educative.io/answers/what-are-response-times-in-ui

Table 6
Error Analysis: Frequency of each error type in terms of proportion (%) of all 43 errors made by our best-performing hybrid model on our test data samples.

Error type	Frequency (%)	Example
Name variations	14	**Mention:** Chopped capsicum **Gold:** pepper **Prediction:** courgette
Accidental lexical similarity	9.3	**Mention:** Sliced pepperoni **Gold:** salami **Prediction:** pepper
Ambiguity	46.5	**Mention:** White wine vinegar **Gold:** vinegar **Prediction:** wine
Insufficient context	11.6	**Mention:** herbs **Gold:** parsley **Prediction:** rosemary
Miscellaneous	18.6	**Mention:** french fries **Gold:** potato **Prediction:** frankfurter
No. of examined errors	43	

each type of approach can be combined to form a superior EL model that balances the trade-offs between performance and inference speed.

Importantly, our work shows that by employing few-shot learning, high accuracy scores (80%–90%) can be obtained for the food EL task, even with fewer than 1000 labelled samples (i.e., mentions linked to their canonical names in our knowledge base). This finding confirms that, by casting EL as a few-shot learning problem, satisfactory performance can be obtained on a specialised domain despite the lack of large amounts of labelled data for model training.

Additionally, our work on producing a new annotated dataset that supports the development and evaluation of EL methods, shows that part of the process, namely the identification of food mentions within ingredient lists, can be done almost fully automatically, with the aid of a transformer-based NER model that was fine-tuned on recipes. This significantly reduced the overhead costs (e.g., time) associated with the prerequisite task of identifying food mentions, and allowed our annotators to focus on the manual EL annotation task itself.

From a practical perspective, our research has several applications. Firstly, our solution has the potential to improve the precision and efficiency of food-related information retrieval systems, such as recipe search engines or food recommendation systems. For instance, by identifying and linking various ingredients mentioned in recipes, a search system can provide more accurate and tailored recipe suggestions to users. Furthermore, academic research in areas such as food nutrition and culinary studies could benefit from our work. An EL model can be used as a tool to automatically extract and analyse food-related data from large volumes of text, e.g., food and nutrition blogs, facilitating the use of such unstructured data in research.

8. Conclusion and future work

In this paper, we describe our work on investigating how deep learning-based approaches that were trained under a few-shot learning setting, can be exploited in the task of EL in the food domain. We also compare such approaches with unsupervised and traditional machine-learning based approaches in terms of performance and inference time. To support the development and evaluation of such approaches, a new food knowledge base, the E.Care KB, was constructed. It contains 664 concepts with 1130 synonyms. Moreover, we developed the E.Care dataset, a novel corpus of 468 cooking recipes in which every unique food name (out of 935 mentions) has been linked to its corresponding concept in the E.Care KB.

The following approaches were developed: (1) two deep learning-based Siamese networks, one underpinned by Bi-LSTMs and the other by sentence transformers; (2) a traditional machine learning-based approach driven by SVMs; and (3) unsupervised approaches based on string similarity algorithms. Combining the strengths of these approaches, we built a hybrid model that obtains optimal performance and inference speed.

Below, we revisit the research questions that we outlined in Section 3 and answer them based on the results of our work:

RQ1: How can state-of-the-art deep learning-based methods be exploited for EL in the food domain, and how well do such methods perform?
As shown by the results of evaluating the Siamese networks that we investigated (Table 2), an effective way to exploit deep learning for food EL is by training Siamese networks to learn similarity between a given food mention and a concept in a knowledge base. Two types of Siamese networks were investigated: one is based on Bi-LSTMs fed with tokens represented using pre-trained fastText embeddings, and the other is underpinned by sentence transformers that make use of sentence-level contextual embedding representations. Although both networks obtained satisfactory accuracy, the most optimal sentence transformer-based Siamese network is more accurate (86.99%) than the Bi-LSTM-based network (82.55%), and is also the fastest in linking mentions (0.004 seconds per mention).

RQ2: How does a deep learning-based approach to food EL compare to previously reported solutions?
Upon comparing our deep learning-based model to our implementations of previously reported approaches such as unsupervised learning based on string similarity algorithms and traditional machine learning-based models such as SVMs, we found that deep learning did not yield the best performance in terms of accuracy. For instance, while our SVM model and cosine similarity-based method respectively obtained 87.19% and 87.43% accuracy, the highest accuracy that our deep learning-based approach obtained is 86.99%, which is definitely competitive yet marginally lower. Its linking time (0.004 seconds per mention) is, however, notably faster than that of the most accurate SVM model (0.66 seconds per mention) and of the cosine similarity-based unsupervised method (0.02 seconds per mention).

RQ3: How can the strengths of different types of approaches be combined to build a solution to the food EL problem that is optimised for both performance and speed?
The strength of our deep learning-based model lies in its capability to effectively filter out unlikely candidates, thus substantially reducing the number of mention-concept pairs whose similarity needs to be assessed. Meanwhile, our cosine similarity-based unsupervised model obtains the best accuracy but by definition does not incorporate semantic information, unlike our SVM model. To combine these strengths into one food EL model, we built a hybrid model that takes the output of the cosine similarity-based model if the similarity between a given mention and a concept is at least 0.95. Otherwise, using the optimised KNN algorithm Annoy, a set of most likely candidate concepts is filtered based on the similarity of their embedding representations – as provided by our chosen sentence transformers – with that of the given mention. Finally, our SVM model assesses the similarity between the given mention and each of the filtered concepts.

We consider our food EL research to be a contribution to the broader area of food data science, which has numerous applications in the domains of nutrition and sustainable consumption. As part of our future work, we plan to expand the E.Care KB with natural-language definitions or descriptions of food concepts, as well as with information on semantic relationships between them (e.g., hierarchical structure). We will then explore how such information can be incorporated as additional information that EL models can leverage.

At the moment, our approaches will always return the best-matching concept for a given mention, and are unable to handle the case where the mention simply does not have any corresponding concept in the knowledge base. Thus, another future direction is the implementation of nil prediction: the sub-task of determining whether a mention cannot be linked to any of the concepts in the knowledge base. We expect that incorporating such as a sub-task should lead to improved accuracy.

CRediT authorship contribution statement

Darius Feher: Methodology, Software, Evaluation, Formal analysis, Writing – original draft and revision. **Faridz Ibrahim:** Software – Knowledge Base, Writing – draft revision. **Zhuyan Cheng:** Data Curation, Writing – draft revision. **Viktor Schlegel:** Conceptualisation, Software – Knowledge base, Writing – draft revision. **Tom Maidment:** Data curation, Writing – draft revision. **James Bagshaw:** Data curation, Writing – draft revision. **Riza Batista-Navarro:** Conceptualisation, Writing – original draft and revision.

Data availability

The data that has been used is confidential.

Acknowledgements

The work described in this paper was partially funded by Nesta, the UK innovation foundation for social good, under the third round of its Collective Intelligence Grants programme. We are also grateful to our student interns, Jingxuan Chen and Yaoxuan Ju, for their help in manually annotating the E.Care dataset.

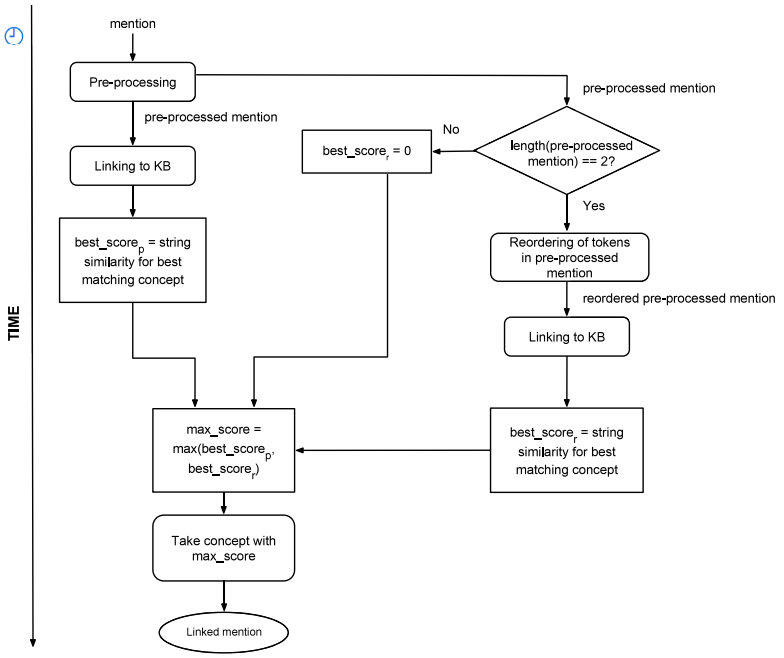
Appendix A. Example annotations

See Table A.1.

Table A.1
Ingredients for "Mushroom Leek Stroganoff", one of the recipes in the E.Care dataset. For each ingredient, the food mentions (as extracted by our named entity recognition model) and the concepts in the E.Care KB that they were manually linked to, are indicated. The original recipe can be found at https://thefeedfeed.com/rainbowplantlife/mushroom-leek-stroganoff.

Recipe ingredient	Food mention	Linked KB concept
2 tablespoons olive oil, divided	olive oil	olive oil
2 large leeks, cleaned, trimmed, and diced	leeks	leek
6 cloves garlic, minced	garlic	garlic
20 ounces mixed mushrooms, sliced	mixed mushrooms	mushroom
1 teaspoon dried thyme, or 1 tablespoon fresh thyme leaves, minced	dried thyme	thyme
	fresh thyme leaves	thyme
1 teaspoon kosher salt, divided and more to taste	kosher salt	kosher salt
1 1/2 cups low-sodium vegetable broth, or water	low-sodium vegetable broth	vegetable stock
	water	water
2 tablespoons tamari, or soy sauce	tamari	soy sauce
	soy sauce	soy sauce
1 tablespoon vegan Worcestershire sauce, optional, omit if gluten free	vegan Worcestershire sauce	worcestershire sauce
1/4 cup all-purpose flour, or gluten-free all-purpose flour	all-purpose flour	wheat flour
	gluten-free all-purpose flour	flour
1/2 cup vegetable stock, or dry white wine	vegetable stock	vegetable stock
	dry white wine	wine
1 (13.5 ounce) can full fat coconut milk	full fat coconut milk	coconut milk
2 tablespoons tahini	tahini	tahini
2 tablespoons nutritional yeast	nutritional yeast	nutritional yeast
1 teaspoon paprika	paprika	paprika
1/2 teaspoon Dijon mustard, or coarse-grain mustard	Dijon mustard	mustard
	coarse-grain mustard	mustard
10–12 ounces pasta, or your favourite grain	pasta	pasta
Fresh parsley, chopped, for garnish, optional	Fresh parsley	parsley
Vegan Parmesan cheese, for garnish, optional	Parmesan cheese	parmesan

Appendix B. Unsupervised string matching-based process for EL



Architecture of the unsupervised string matching approach. Key: $best\_score_p$: best score obtained for the pre-processed mention; $best\_score_r$: best score obtained for the reordered pre-processed mention.

Appendix C. Evaluation results for unsupervised EL approach including synonyms

See Table C.1.

Table C.1
Evaluation results for string matching methods including synonyms in the knowledge base. A similarity threshold of 0.9 was used and the Jaro index was applied for token removal. Best results are highlighted in bold.

Algorithm	Accuracy (%)	Time (sec/mention)
Jaccard	68.22	0.017
LCS	70.19	0.016
EDIT	73.39	**0.013**
Jaro	79.80	0.09
Jaro–Winkler	81.52	0.09
Novelty Bigram	84.48	0.77
Q-gram	**85.22**	0.12
Cosine	**85.22**	0.07

References

Alokaili, A., & Menai, M. E. B. (2020). SVM ensembles for named entity disambiguation. *Computing*, *102*(4), 1051–1076. http://dx.doi.org/10.1007/s00607-019-00748-x.

Arighi, C., Hirschman, L., Lemberger, T., Bayer, S., Liechti, R., Comeau, D., et al. (2017). Bio-ID track overview. In *Proc. biocreative workshop, vol. 482* (p. 376).

Basaldella, M., Liu, F., Shareghi, E., & Collier, N. (2020). COMETA: A corpus for medical entity linking in the social media. http://dx.doi.org/10.48550/arXiv.2010.03295.

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. http://dx.doi.org/10.1162/tacl_a_00051.

Ceccarelli, D., Lucchese, C., Orlando, S., Perego, R., & Trani, S. (2013). Learning relatedness measures for entity linking. In *Proceedings of the 22nd ACM international conference on information & knowledge management* (pp. 139–148). http://dx.doi.org/10.1145/2505515.2505711.

Cenikj, G., Popovski, G., Stojanov, R., Koroušić Seljak, B., & Eftimov, T. (2020). BuTTER: BidirecTional LSTM for food named-entity recognition. In *2020 IEEE international conference on big data* (pp. 3550–3556). IEEE, http://dx.doi.org/10.1109/BigData50022.2020.9378151.

Chakraborty, S., Raj, H., Gureja, S., Jain, T., Hassan, A., & Basu, S. (2023). Evaluating the robustness of biomedical concept normalization. In *Transfer learning for natural language processing workshop* (pp. 63–73). PMLR.

Chong, W. H., & Lim, E. P. (2018). Implicit linking of food entities in social media. In *Joint European conference on machine learning and knowledge discovery in databases* (pp. 169–185). Springer, http://dx.doi.org/10.1007/978-3-030-10997-4_11.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and psychological measurement*, *20*(1), 37–46.

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., & Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. http://dx.doi.org/10.48550/arXiv.1705.02364.

De Cao, N., Izacard, G., Riedel, S., & Petroni, F. (2020). Autoregressive entity retrieval. http://dx.doi.org/10.48550/arXiv.2010.00904.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. http://dx.doi.org/10.48550/arXiv.1810.04805.

Dong, X., & Shen, J. (2018). Triplet loss in siamese network for object tracking. In *Proceedings of the European conference on computer vision* ECCV, (pp. 459–474).

Dooley, D. M., Griffiths, E. J., Gosal, G. S., Buttigieg, P. L., Hoehndorf, R., Lange, M. C., et al. (2018). FoodOn: a harmonized food ontology to increase global food traceability, quality control and data integration. *npj Science of Food*, *2*(1), 1–10.

Eftimov, T., Korošec, P., & Koroušić Seljak, B. (2017). StandFood: standardization of foods using a semi-automatic system for classifying and describing foods according to FoodEx2. *Nutrients*, *9*(6), 542. http://dx.doi.org/10.3390/nu9060542.

El Vaigh, C. B., Goasdoué, F., Gravier, G., & Sébillot, P. (2019). Using knowledge base semantics in context-aware entity linking. In *Proceedings of the ACM symposium on document engineering 2019* (pp. 1–10). http://dx.doi.org/10.1145/3342558.3345393.

El Vaigh, C. B., Torregrossa, F., Allesiardo, R., Gravier, G., & Sébillot, P. (2020). A correlation-based entity embedding approach for robust entity linking. In *2020 IEEE 32nd international conference on tools with artificial intelligence* (pp. 949–954). IEEE, http://dx.doi.org/10.1109/ICTAI50040.2020.00148.

Fakhraei, S., Mathew, J., & Ambite, J. L. (2019). NSEEN: Neural semantic embedding for entity normalization. In *Joint European conference on machine learning and knowledge discovery in databases* (pp. 665–680). Springer, http://dx.doi.org/10.1007/978-3-030-46147-8_40.

Fears, R., Canales, C., Ter Meulen, V., & von Braun, J. (2019). Transforming food systems to deliver healthy, sustainable diets—the view from the world's science academies. *The Lancet Planetary Health*, *3*(4), e163–e165. http://dx.doi.org/10.1016/S2542-5196(19)30038-5.

Ha, T. T., Nguyen, V. N., Nguyen, K. H., Nguyen, K. A., & Than, Q. K. (2021). Utilizing sbert for finding similar questions in community question answering. In *2021 13th international conference on knowledge and systems engineering* (pp. 1–6). IEEE, http://dx.doi.org/10.1109/KSE53942.2021.9648830.

Hamon, T., Tabanou, V., Mougin, F., Grabar, N., & Thiessard, F. (2017). POMELO: Medline corpus with manually annotated food-drug interactions. In *Proceedings of the biomedical NLP workshop associated with RANLP 2017* (pp. 73–80). http://dx.doi.org/10.26615/978-954-452-044-1_010.

Harrington, R. A., Adhikari, V., Rayner, M., & Scarborough, P. (2019). Nutrient composition databases in the age of big data: foodDB, a comprehensive, real-time database infrastructure. *BMJ Open*, *9*(6), Article e026652. http://dx.doi.org/10.1136/bmjopen-2018-026652.

Hosseini, H., & Bagheri, E. (2021). Learning to rank implicit entities on Twitter. *Information Processing & Management*, *58*(3), Article 102503. http://dx.doi.org/10.1016/j.ipm.2021.102503.

Hosseini, H., Nguyen, T. T., Wu, J., & Bagheri, E. (2019). Implicit entity linking in tweets: An ad-hoc retrieval approach. *Applied Ontology*, *14*(4), 451–477. http://dx.doi.org/10.3233/AO-190215.

Irrera, O., & Silvello, G. (2021). Background linking: Joining entity linking with learning to rank models. In *IRCDL* (pp. 64–77).

ITMO University (2016). Food product ontology. URL: https://vest.agrisemantics.org/content/food-product-ontology. (Accessed 12 November 2022).

Jia, B., Wu, Z., Zhou, P., & Wu, B. (2021). Entity linking based on sentence representation. *Complexity*, *2021*, http://dx.doi.org/10.1155/2021/8895742.

João, R. S., Fafalios, P., & Dietze, S. (2019). Same but different: distant supervision for predicting and understanding entity linking difficulty. In *Proceedings of the 34th ACM/SIGAPP symposium on applied computing* (pp. 1019–1026). http://dx.doi.org/10.1145/3297280.3297381.

Jonquet, C., Shah, N., Youn, C., Callendar, C., Storey, M. A., & Musen, M. (2009). NCBO annotator: semantic annotation of biomedical data. In *International semantic web conference, poster and demo session, vol. 110*.

Karadeniz, I., & Özgür, A. (2019). Linking entities through an ontology using word embeddings and syntactic re-ranking. *BMC Bioinformatics*, *20*, 1–12. http://dx.doi.org/10.1186/s12859-019-2678-8.

Kaur, A. (2015). A novel approach for syntactic similarity between two short text. *International Journal of Scientific & Technology Research*, *4*(06), 2277–8616.

Klie, J. C., de Castilho, R. E., & Gurevych, I. (2020). From zero to hero: Human-in-the-loop entity linking in low resource domains. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 6982–6993). http://dx.doi.org/10.18653/v1/2020.acl-main.624.

Landis, J. R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, *33*(1), 159–174. http://dx.doi.org/10.2307/2529310.

Laskar, M. T. R., Chen, C., Martsinovich, A., Johnston, J., Fu, X. Y., Tn, S. B., et al. (2022). BLINK with elasticsearch for efficient entity linking in business conversations. http://dx.doi.org/10.18653/v1/2022.naacl-industry.38.

Liang, Z., & Shen, J. (2019). Local semantic siamese networks for fast tracking. *IEEE Transactions on Image Processing*, *29*, 3351–3364. http://dx.doi.org/10.1109/TIP.2019.2959256.

LIRMM (2013). Open food facts food ontology. URL: https://vest.agrisemantics.org/content/open-food-facts-food-ontology. (Accessed 12 November 2022).

Ma, P., Zhang, Z., Li, Y., Yu, N., Sheng, J., Küçük-McGinty, H., et al. (2022). Deep learning accurately predicts food categories and nutrients based on ingredient statements. *Food Chemistry*, *391*, Article 133243. http://dx.doi.org/10.1016/j.foodchem.2022.133243.

Marin, J., Biswas, A., Ofli, F., Hynes, N., Salvador, A., Aytar, Y., et al. (2019). Recipe1M+: A dataset for learning cross-modal embeddings for cooking recipes and food images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Martin, L., Muller, B., Ortiz Suárez, P. J., Dupont, Y., Romary, L., de la Clergerie, É., et al. (2020). CamemBERT: a tasty french language model. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 7203–7219). Online: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2020.acl-main.645.

Miller, V., Singh, G. M., Onopa, J., Reedy, J., Shi, P., Zhang, J., et al. (2021). Global dietary database 2017: data availability and gaps on 54 major foods, beverages and nutrients among 5.6 million children and adults from 1220 surveys worldwide. *BMJ Global Health*, *6*(2), Article e003585. http://dx.doi.org/10.1136/bmjgh-2020-003585.

Minaee, S., & Liu, Z. (2017). Automatic question-answering using a deep similarity neural network. In *2017 IEEE global conference on signal and information processing* (pp. 923–927). IEEE, http://dx.doi.org/10.1109/GlobalSIP.2017.8309095.

Mohammadi, E., Naji, N., Marceau, L., Queudot, M., Charton, E., Kosseim, L., et al. (2020). Cooking up a neural-based model for recipe classification. In *Proceedings of the twelfth language resources and evaluation conference* (pp. 5000–5009).

Morning Consult (2022). Consumers are avid recipe users: Understanding recipe and non-recipe occasions. URL: https://morningconsult.com/2022/04/18/consumers-avid-recipe-users/. (Accessed 4 November 2022).

Moro, A., Raganato, A., & Navigli, R. (2014). Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, *2*, 231–244. http://dx.doi.org/10.1162/tacl_a_00179.

Neculoiu, P., Versteegh, M., & Rotaru, M. (2016). Learning text similarity with siamese recurrent networks. In *Proceedings of the 1st workshop on representation learning for NLP* (pp. 148–157). http://dx.doi.org/10.18653/v1/W16-1617.

Ninomiya, A., & Ozaki, T. (2020). Cooking recipe analysis based on sequences of distributed representation on procedure texts and associated images. In *Proceedings of the 12th workshop on multimedia for cooking and eating activities* (pp. 13–18). http://dx.doi.org/10.1145/3379175.3391710.

Nozza, D., Sas, C., Fersini, E., & Messina, E. (2019). Word embeddings for unsupervised named entity linking. In *International conference on knowledge science, engineering and management* (pp. 115–132). Springer, http://dx.doi.org/10.1007/978-3-030-29563-9_13.

Papantoniou, K., Efthymiou, V., & Flouris, G. (2021). EL-NEL: Entity linking for greek news articles. In *ISWC (Posters/Demos/Industry)*.

Parravicini, A., Patra, R., Bartolini, D. B., & Santambrogio, M. D. (2019). Fast and accurate entity linking via graph embedding. In *Proceedings of the 2nd joint international workshop on graph data management experiences & systems (GRADES) and network data analytics* (NDA), (pp. 1–9). http://dx.doi.org/10.1145/3327964.3328499.

Popovski, G., Kochev, S., Koroušić Seljak, B., & Eftimov, T. (2019a). FoodIE: A rule-based named-entity recognition method for food information extraction. In *ICPRAM* (pp. 915–922). http://dx.doi.org/10.5220/0007686309150922.

Popovski, G., Koroušić Seljak, B., & Eftimov, T. (2019b). FoodOntoMap: Linking food concepts across different food ontologies.. In *KEOD* (pp. 195–202). http://dx.doi.org/10.5220/0008353201950202.

Popovski, G., Koroušić Seljak, B., & Eftimov, T. (2019c). FoodBase corpus: a new resource of annotated food entities. *Database*, *2019*, http://dx.doi.org/10.1093/database/baz121.

Public Health England (2015). Composition of foods integrated dataset (CoFID). URL: https://www.gov.uk/government/publications/composition-of-foods-integrated-dataset-cofid. (Accessed 12 November 2022).

Ramachandra, B., Jones, M., & Vatsavai, R. (2020). Learning a distance function with a siamese network to localize anomalies in videos. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision* (pp. 2598–2607).

Ravi, M. P. K., Singh, K., Mulang', I. O., Shekarpour, S., Hoffart, J., & Lehmann, J. (2021). CHOLAN: A modular approach for neural entity linking on wikipedia and wikidata. http://dx.doi.org/10.48550/arXiv.2101.09969.

Recchia, G., & Louwerse, M. (2013). A comparison of string similarity measures for toponym matching. http://dx.doi.org/10.1145/2534848.2534850.

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. http://dx.doi.org/10.18653/v1/D19-1410.

Shanaz, A. L. F., & Ragel, R. G. (2021). Wikidata based person entity linking in news articles. In *2021 10th international conference on information and automation for sustainability* (pp. 66–70). IEEE, http://dx.doi.org/10.1109/ICIAfS52090.2021.9606139.

Shih, C. H., Yan, B. C., Liu, S. H., & Chen, B. (2017). Investigating siamese LSTM networks for text categorization. In *2017 Asia-Pacific signal and information processing association annual summit and conference* (APSIPA ASC), (pp. 641–646). IEEE, http://dx.doi.org/10.1109/APSIPA.2017.8282104.

Silva, N., Ribeiro, D., & Ferreira, L. (2019). Information extraction from unstructured recipe data. In *Proceedings of the 2019 5th international conference on computer and technology applications* (pp. 165–168). http://dx.doi.org/10.1145/3323933.3324084.

Song, L., Gong, D., Li, Z., Liu, C., & Liu, W. (2019). Occlusion robust face recognition based on mask learning with pairwise differential siamese network. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 773–782). http://dx.doi.org/10.48550/arXiv.1908.06290.

Spitters, M., Bonnema, R., Rotaru, M., & Zavrel, J. (2010). Bootstrapping information extraction mappings by similarity-based reuse of taxonomies. In *CEUR Workshop Proceedings, vol. 673*.

Stojanov, R., Popovski, G., Cenikj, G., Koroušić Seljak, B., Eftimov, T., et al. (2021). A fine-tuned bidirectional encoder representations from transformers model for food named-entity recognition: Algorithm development and validation. *Journal of Medical Internet Research*, *23*(8), Article e28229. http://dx.doi.org/10.2196/28229.

Syed, M. H., & Chung, S. T. (2021). MenuNER: Domain-adapted BERT based NER approach for a domain with limited dataset and its application to food menu domain. *Applied Sciences*, *11*(13), 6007. http://dx.doi.org/10.3390/app11136007.

Torfi, A., Shirvani, R. A., Keneshloo, Y., Tavaf, N., & Fox, E. A. (2020). Natural language processing advancements by deep learning: A survey. http://dx.doi.org/10.48550/arXiv.2003.01200.

Tsai, C. T., & Roth, D. (2016). Cross-lingual wikification using multilingual embeddings. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Human language technologies* (pp. 589–598). http://dx.doi.org/10.18653/v1/N16-1072.

University of Glasgow (2015). SAMUELS project. URL: https://www.gla.ac.uk/schools/critical/research/fundedresearchprojects/samuels/outputs/#toolsandcorpora. (Accessed 6 November 2022).

US Department of Agriculture, Agricultural Research Service (2019). USDA food and nutrient database for dietary studies. URL: https://data.nal.usda.gov/dataset/food-and-nutrient-database-dietary-studies-fndds. (Accessed 12 May 2023).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Ventirozos, F., Jacobo-Romero, M., Clinch, S., & Batista-Navarro, R. (2021). Interactive clustering of cooking recipe instructions: Towards the automatic detection of events involving kitchen devices. In *2021 IEEE 15th international conference on semantic computing* (pp. 341–346). IEEE, http://dx.doi.org/10.1109/ICSC50631.2021.00064.

Wang, Y., Qin, J., & Wang, W. (2017). Efficient approximate entity matching using jaro-winkler distance. In *International conference on web information systems engineering* (pp. 231–239). Springer, http://dx.doi.org/10.1007/978-3-319-68783-4_16.

Willett, W., Rockström, J., Loken, B., Springmann, M., Lang, T., Vermeulen, S., et al. (2019). Food in the Anthropocene: the EAT–Lancet Commission on healthy diets from sustainable food systems. *The Lancet*, *393*(10170), 447–492. http://dx.doi.org/10.1016/S0140-6736(18)31788-4.

Wu, G., He, Y., & Hu, X. (2018). Entity linking: an issue to extract corresponding entity with knowledge base. *IEEE Access*, *6*, 6220–6231. http://dx.doi.org/10.1109/ACCESS.2017.2787787.

Yamakata, Y., Mori, S., & Carroll, J. A. (2020). English recipe flow graph corpus. In *Proceedings of the 12th language resources and evaluation conference* (pp. 5187–5194).

Yuan, H., Yuan, Z., & Yu, S. (2022). Generative biomedical entity linking via knowledge base-guided pre-training and synonyms-aware fine-tuning. http://dx.doi.org/10.48550/arXiv.2204.05164.

Zhang, S., Cheng, H., Vashishth, S., Wong, C., Xiao, J., Liu, X., et al. (2022). Knowledge-rich self-supervision for biomedical entity linking. In *Findings of the association for computational linguistics* (pp. 868–880).

Zhang, W., Hua, W., & Stratos, K. (2021). Entqa: Entity linking as question answering. http://dx.doi.org/10.48550/arXiv.2110.02369.

Zhang, W., Sim, Y. C., Su, J., & Tan, C. L. (2011). Entity linking with effective acronym expansion, instance selection and topic modeling. In *Twenty-second international joint conference on artificial intelligence*. http://dx.doi.org/10.5591/978-1-57735-516-8/IJCAI11-319.

Zheng, J. G., Howsmon, D., Zhang, B., Hahn, J., McGuinness, D., Hendler, J., et al. (2015). Entity linking for biomedical literature. *BMC Medical Informatics and Decision Making*, *15*(1), 1–9. http://dx.doi.org/10.1186/1472-6947-15-S1-S4.