Computing (2020) 102:1323–1344
https://doi.org/10.1007/s00607-020-00792-y

Recommendations from cold starts in big data

David Ralph[1] · **Yunjia Li**[2] · **Gary Wills**[1] · **Nicolas G. Green**[1]

Received: 4 September 2019 / Accepted: 20 January 2020 / Published online: 29 January 2020

Abstract
This paper examines the challenging problem of new user cold starts in subset labelled and extremely sparsely labelled big data. We introduce a new Isle of Wight Supply Chain (IWSC) dataset demonstrating these characteristics. We also introduce a new technique addressing these challenges, the Transitive Semantic Relationships (TSR) model, which infers potential relationships from user and item text content and few labelled examples. We perform both implicit and explicit evaluation of TSR as a recommender system and from new user cold starts we achieve a hit-rate@10 of 77% on a collection of 630 items with only 376 supply-chain consumer labels, and 67% with only 142 supply-chain supplier labels, demonstrating a high level of performance even with extremely few labels in challenging cold-start scenarios. TSR is suitable for any dataset featuring few labels and user and item content, where similarity of content indicates similar relationship forming capability. TSR can be used as a standalone recommender system or to complement existing high-performance recommender models that require more labels or do not support cold starts.

Keywords Recommender systems · Information retrieval · Data mining · Sparse data · Partially labelled data

Mathematics Subject Classification 68P20 · 68T50

✉ David Ralph
D.Ralph@ecs.soton.ac.uk

Yunjia Li
yunjia.li@launchbase.solutions

Gary Wills
GBW@ecs.soton.ac.uk

Nicolas G. Green
NG2@ecs.soton.ac.uk

1 Electronics and Computer Sciences (ECS), University of Southampton, University Rd, Southampton SO17 1BJ, UK

2 Launch International LTD, 3000a Parkway, Whiteley, Fareham PO15 7FX, UK

1 Introduction

New Big Data recommendation systems face a high barrier to entry due to the large labelled data requirement of most existing recommendation techniques such as collaborative filtering and bespoke deep learning models such as Suglia et al. [23]. Obtaining this labelled data, such as user interactions or human judgements, is particularly problematic in highly specialised or commercially competitive domains where this labelling may not yet exist or not be freely available, often requiring an expensive expert or crowd-sourced labelling. As such, techniques that function well with few labels are highly desirable.

In this paper we investigate this data problem and the limitations of existing recommender systems, and go on to introduce a new technique for providing personalised recommendations for new users even in highly challenging datasets where few labels are available for most items, and no labels are known for the new user, and without the need for the user to answer a questionnaire. We achieve this by using user and item content information in the form of natural language descriptive text to expand on the few or sparsely distributed known relationships in the dataset.

2 Background

Recommendation systems, like search systems, aim to identify the most relevant items from a collection to fulfil a user's search intent. In the case of recommendation systems, by using known information about the user and items, including either interaction data, content data, or both, to produce personalised results, typically without requiring a query. In contrast to query driven search systems, this means the user is not required to articulate their search intent, which allows recommendations to be made without direct involvement from the user.

Collaborative recommender systems aggregate user interactions to find similar users and recommend the items they liked. Common techniques include collaborative filtering, where matrix factorisation is used to reduce the dimensionality of the sparse matrix of user item interactions. The resulting dense matrix can be used to recommend items based on a given user's past interactions [15]. However, due to dependence on user interactions, collaborative approaches present issues when items are time sensitive or competitive as items may not remain valid long enough to accumulate a significant user record [27]. Further, this approach can result in positive feedback loops where items with more numerous or diverse interaction histories are more frequently shown to users; this virality effect can result in a few generic or broadly applicable documents being disproportionately recommended, while newer are not promoted due to less existent user behaviour data [27].

In contrast, content-based recommender systems recommend items based on similarity to content information known about items a user has previously interacted with. In recent works such as Suglia et al. [23] and Musto et al. [18] description embeddings are used for this comparison. A common approach used in both papers is to generate a representation for the user by averaging the description embeddings of the items the user has interacted with previously. Content-based systems are less dependent on items

having extensive interaction histories as they can recommend new items based purely on content similarity, but they still require the user to have known past interactions to use for comparison.

Neither of these approaches can be used for user-wise cold starts, where there is no behaviour record for the user. These approaches also depend on users seeking documents similar to previous searches, which may not be true between sessions; that is, if a user changes their search objective the data may no longer be relevant because the nature of the relationship (the user's need) has changed [13].

2.1 Sparsity and cold-starts

The cost of labelling data is highly dependent on the complexity of the task, specifically the time needed per human annotation and the expertise required. Snow et al. [22] find that for tasks such as textual entailment and word sense disambiguation approximately four non-expert labels have similar quality to one expert label. Grady and Lease [7] investigate crowdsourcing binary relevance labelling tasks and find that tasks where annotators must use item descriptions achieve poorer accuracy and require greater time per judgement than tasks using titles.

In some cases, datasets may be too large for comprehensive manual labelling and may only be viable to label by observing user behaviour, which requires a system able to function with very few labels without exclusively preferring the already labelled subset of the data. Such systems could be used to bootstrap a recommendation platform where user interactions can then be observed to enhance the model or train an alternative model which performs well with many labels. This is also related to the cold-start problem where newly added items have no past interaction data, such as in high velocity big data.

The cold-start problem is typically divided into the two sub-problems of item-wise (new item) and user-wise (new user) cold starts. The item-wise case is commonly addressed by content-based and hybrid recommender systems; however, the user-wise case has received less attention, even in scenarios where content information for the user is available.

Content based and hybrid recommender systems reduce the requirement for item labels by making use of item content, such as descriptions. Many such systems rely on either knowledge bases and ontologies [28], which do not avert the requirement of experts for new or commercially guarded domains, or tags and categorisation [25], which requires either many labels or distinct groupings in the data.

Yuan et al. [27] examine the real-world data problem of matching users to job postings, where items are time sensitive and new items are very frequent. They make the case that high performance techniques that require item labels can be generalised to cold-start items by pairing labelled and unlabelled items based on the similarity of their content.

In this work, we introduce a novel technique to address both user-wise and item-wise cold starts using user and item content and minimal labels. This paper focuses mainly on user-wise cold starts and the data sparsity problem as they have received less attention in the literature.

2.2 Provenance and visualisation

Existing search and recommender systems that make use of deep learning or statistical methods such as matrix factorisation typically output only a series of ranked items or confidence scores in response to a query, and the rationale behind these decisions in unknown.

In the case of scores from neural networks or other learning-to-rank models the reasoning behind the algorithm's decision is unknowable as it is derived from the free parameters of the model which only have meaning inside the model and cannot be used to meaningfully explain results, these are referred to as "Black Box" systems [29]. Much research has been done into approaches for understanding the internals of deep learning models via visualisation, particularly in the areas of text summarisation [21] and computer vision [26,30], and some works have looked at understanding the language models used to generate text embeddings for content-based item comparisons [16]. However, while these visualisation techniques offer some insights into the factors the model considers important, they cannot produce a reasoned explanation for the response to individual queries in any way similar to how a human decision maker might evidence their decision.

In contrast to this are "White Box" systems which produce meaningful provenance that can be used to explain results and study the operation of the model, these typically include rule based models, and expert systems. Herlocker et al. [10] discuss how in user-facing scenarios some techniques such as collaborative filtering can be presented as either a White Box or Black Box model, by giving feedback to the users based on either the operational steps of the model (White Box), or the inputs and outputs of the system such as user evaluations of the quality of results (Black Box).

Detailed provenance data such as lists of decision making steps, inferences, and knowledge and items considered when evaluating a query can be used to produce visualisations such as graphical plots or flow diagrams to help users understand the reasoning behind a result, increasing their confidence in the decision, or highlighting potential flaws in the model. This makes provenance highly desirable both during development for the purposes of debugging and improving the model, and for user facing systems as users have greater trust in answers that are explainable and can make more informed decisions based on the results [10].

3 Isle of Wight supply chain dataset

We examine the case of supply chain on the Isle of Wight. We introduce a new dataset for this task, which we name the Isle of Wight Supply Chain (IWSC) dataset. The data consists of varying length text descriptions of 630 companies on the Isle of Wight taken via web scraping from the websites of IWChamber [1], IWTechnology [2], and Marine Southeast [3].

HTML tags and formatting have been removed, but the descriptions are otherwise unaltered and are provided untokenized, without substitutions, and complete with punctuation. Some descriptions contain product codes, proper nouns, and other non-dictionary words. The descriptions are typically a few sentences describing the market

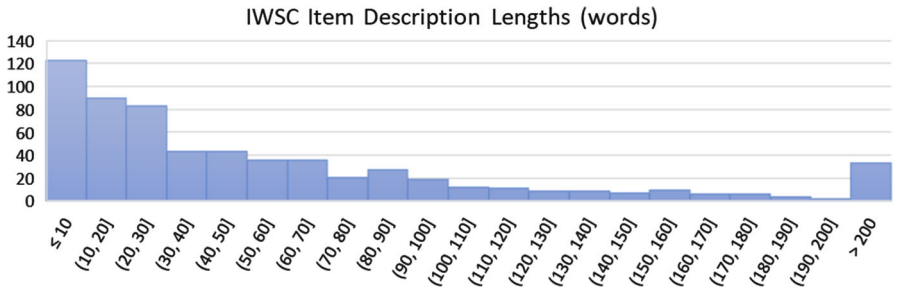


Fig. 1 Histogram of item description lengths in the IWSC dataset

Table 1 Labels in the IWSC dataset

Label name	Total labels	Labelled items	Unique targets
SL_suppliers	142	15	75
SL_not_suppliers	563	16	120
SL_consumers	376	17	117
SL_not_consumers	712	16	157
SL_competitors	82	15	49
SL_not_competitors	396	17	99
ES_suppliers	92	48	76
ES_consumers	207	51	171
ES_competitors	95	53	82
ES_unrelated	431	75	299

role of the company, or a general description of the company's activities or products. Some of the descriptions also contain a list of keywords, but this is included as part of the descriptive text and not as an isolated feature. The mean description length is 61 words, or 412 characters (including whitespace). The distribution of description lengths is shown in Fig. 1.

The IWSC dataset is provided with two discrete sets of labels intended to evaluate algorithmic performance in different scenarios. In both cases, the labels are binary, directed, human judgements of market relatedness based on the company descriptions. The number and distribution of labels is shown in Table 1. These labels are speculative potential relationships, not necessarily real existing relationships. We choose to provide binary labels as real-world supply chain relationships are typically multi-class binary relationships. i.e. any two companies either are or are not in each possible type of supply chain relationship.

The first label set, IWSC-SL, consists of the labels in Table 1 prefixed "SL". These labels are concentrated on a small number of labelled items, relating them to a random distribution of other items (both labelled and unlabelled). These labels are intended for evaluation in the case that we only have records for a small subset of items and must extrapolate from this to perform inferences on many unseen items. We refer to this scenario as "Subset Labelling" (SL).

The second label set, IWSC-ES, consists of the labels in Table 1 prefixed "ES". The labels are randomly distributed across all items with no intentional patterns (random pairs were selected for labelling). These labels are intended for evaluation in the case that known items have very few labels and many are entirely unlabelled, in contrast to common recommender system datasets such as Movie Reviews (MR) [19], Customer Reviews (CR) [11], and MovieLens [8], where most items have many recorded interactions. While in those examples the labels are sparse as most possible item pairs are unlabelled, in our scenario, which we refer to as "Extremely Sparse" (ES) labelling, there is the additional condition that many items in the dataset do not occur in any of these pairs.

Figures 2 and 3 illustrate these two different label distributions. Both figures show the same plot of IWSC items, arranged using a two dimensional t-SNE [17] reduction of the 512 dimensional document embeddings generated from each item's descriptive text using Universal Sentence Encoder (USE) [6]. The plots are then annotated with the labels from IWSC-SL and IWSC-ES respectively. The intention of these figures is to illustrate the difference in the connectivity between items in the subset labelling (few labelled items) and extremely sparse labelling (few labels per item) scenarios.

For the problem of effective recommendations from few labels, we set the four following tasks:

1. Recommendation of consumers using IWSC-SL labels and item descriptions
2. Recommendation of suppliers using IWSC-SL labels and item descriptions
3. Recommendation of consumers using IWSC-ES labels and item descriptions
4. Recommendation of suppliers using IWSC-ES labels and item descriptions

Networks of user-to-user relationships can be also represented as using a multi-layer approach, where each layer shows relationships of a particular type [4]. This may be suitable for the supply chain scenario, where multiple relationship types could be used to give supporting evidence for the likelihood of a predicted relationship, such as considering known competitors when predicting potential suppliers or consumers. These tasks could also be expressed as two multi-class classification problems (one each for IWSC-SL and IWSC-ES). However, in this paper, we focus on the case where only one relationship type is known, as it is the more general case in recommender systems, and in particular we address four single-class recommendation tasks set out above.

This dataset has been made publicly available. Section 8 provides more information on the repository.

4 Transitive semantic relationships

We introduce a novel approach to approach the problems of extremely sparse labelling and subset labelling previously described, that we call "Transitive Semantic Relationships" (TSR). TSR uses item content for unsupervised comparison of items to expand the coverage of the few available labels. This is conceptually similar to other hybrid recommenders making use of content such as Vuurens et al. [24] and He et al. [9], but we implement a novel approach using user and item content embeddings and inferen-

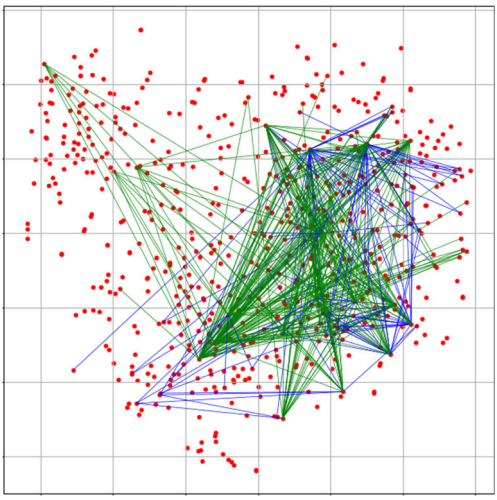


Fig. 2 A 2D t-SNE plot of IWSC item description embeddings showing labels for the SL tasks. This is an example of "Subset Labelling", where all known labels are distributed over a small subset of items in a big dataset. Each item in the labelled subset has several labels, but all other items are completely unlabelled

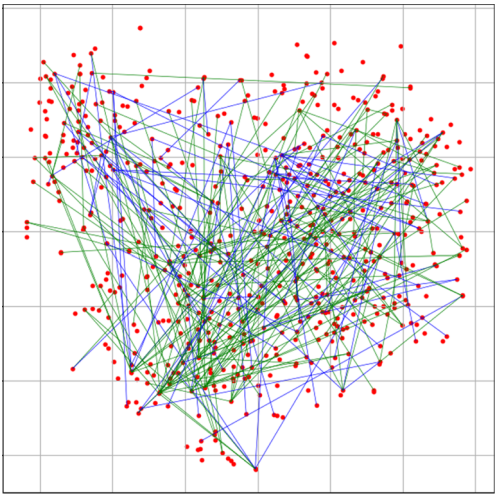


Fig. 3 A 2D t-SNE plot of IWSC item description embeddings showing labels for the ES tasks. This is an example of "Extremely Sparse Labelling" where a small number of labels are distributed randomly across a big dataset. The labelled items each have very few labels and some items are completely unlabelled

tial logic instead of learned or averaged user embeddings. Our work is also similar in principle to the approach in Yuan et al. [27] where unlabelled and labelled items are paired based on content similarity, but while that work focuses on methods for finding the most similar historic items and then using the paired item's score for the new item, we instead go on to define methods by which recommendation scores can be calculated using the semantic difference between the new item and one or more labelled items and also define a combined semantic difference approach for cases where both the query and target are unlabelled. As such our approach is suitable for both user-wise and item-wise cold-starts. These differences make our approach suitable for datasets with fewer labels and covers edge cases such as double cold starts. TSR also has the benefits of producing provenance that is both intuitively understandable and easy to visualise;

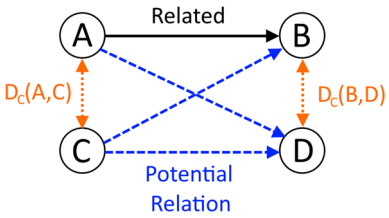


Fig. 4 Illustration of Transitive Semantic Relationships. The dotted lines labelled $D_C(A, C)$ and $D_C(B, D)$ represent the cosine distance between the content embeddings of items A and C, and B and D respectively

supporting both partitioned (user-to-item) and contiguous (user-to-user/item-to-item) datasets; and not requiring a computationally expensive training process.

4.1 Theory

Transitive Semantic Relationships are based on an apparent transitivity property of many types of data items, where it is the case that items which are described similarly are likely to have similar relationships to other data items. Take for example, the supply chain: if company A, a steel mill and company B, a construction firm are known to have the relationship A supplies (sells to) B, it is likely that some other companies C, another steel mill, and D, another construction firm, would have a similar relationship. Given content information about each company, such as a text description of their product or market role, and the example relationship $A \rightarrow B$, we can infer the potential relationships $C \rightarrow D$, $A \rightarrow D$, and $C \rightarrow B$. We illustrate this example in Fig. 4.

It follows that the greater the similarity between an item of interest and an item in a known relationship, the greater confidence we can have that the relationship is applicable. Given some fixed length vector representation of the content information for each item, we can use cosine similarity to measure similarity between the items. The vector representation should ideally capture semantic features of the auxiliary information that indicate whether the items they describe are similar in function in terms of the known relationship. If the vector representations fulfil this criterion, then the cosine similarity between two items is their semantic similarity. It then follows that we can determine the confidence that some query item and some target item share a relationship by measuring the cosine similarity of the semantic vectors for the query and the target with another pair of items that are known to share a relationship of the type of interest.

We herein use the cosine distance rather than the similarity as we consider it easier to interpret when results are visualised and when distance values are weighted. Cosine distance values range from 0 (completely similar) to 1 (no similarity), so to keep scores in the range 0 to 1 when combining the two distances of the query and the target from the labelled pair, we take the sum of the distances over 2. To obtain a confidence value where 1 is full confidence of applicability and 0 is no confidence we subtract the combined distance from 1. We refer to this scoring metric as the combined-cosine-distance, or more generally the combined-semantic-distance.

Continuing from the prior example illustrated in Fig. 4, if the cosine distance of A and C is $D_C(A, C)$, and the distance of B and D is $D_C(B, D)$, we can calculate the confidence for each inferred relationship to be as shown in Eqs. (1), (2), and (3).

We include the $+0$ in Eqs. (1) and (2) for consistency to represent $D_C(A, A)$ for the former and $D_C(B, B)$ for the later but the cosine distance between an item and itself is always 0.

$$A \rightarrow D = 1 - \frac{0 + D_C(B, D)}{2} \tag{1}$$

$$C \rightarrow B = 1 - \frac{D_C(A, C) + 0}{2} \tag{2}$$

$$C \rightarrow D = 1 - \frac{D_C(A, C) + D_C(B, D)}{2} \tag{3}$$

To further illustrate this, if C is very similar to A, for example $let\ D_C(A, C) = 0.2$, but D was only slightly similar to B, $let\ D_C(B, D) = 0.8$, then we can calculate: $A \rightarrow D = 0.6$, $C \rightarrow B = 0.9$, $C \rightarrow D = 0.5$, indicating that there is a good chance that C could share a similar relationship with B as A does, but other new relations are unlikely. In another example, if C remains similar to A, $let\ D_C(A, C) = 0.2$, but we make D more similar to B, $let\ D_C(B, D) = 0.3$, then we calculate: $A \rightarrow D = 0.85$, $C \rightarrow B = 0.9, C \rightarrow D = 0.75$, showing that while all relationships are likely, higher confidence scores are awarded when there is less uncertainty due to dissimilarity with the labelled pair.

4.2 Application

The previous scenarios suppose that we have already pre-determined the items of interest for comparison. However, we can extend this principle to selection of items for comparison, given an input item to use as a query. Note that this is not a query in the sense of traditional search engines but is content information for an item for which we want to find relations (e.g. a user or item description).

First, we must make the distinction between cases where relationships map from one space to some other non-overlapping space, for example separate document collections, and the alternative case where items on either side of the relationship co-exist in the same space. A practical example of the former might be a collection of resumes and a collection of job adverts, while an example of the later might be descriptions of companies looking for supply chain opportunities, as in the IWSC dataset on which we evaluate TSR later in this paper. The TSR scoring does not differentiate between these two dataset types, but in the former case, with separate item collections, it is only necessary to make similarity comparisons between items in the same collection and irrespective of the total number of collections, we need only examine the collections featuring items on either end of at least one example of the relationship type of interest; this may be a useful filtering criteria in datasets featuring many types of relationships across many non-overlapping collections.

Having identified the collections that are of interest, we can optionally apply additional filtering of items before similarity comparison, such as by using item meta-data or additional auxiliary information, for example, only considering recent information, or limiting by language or region. This filtering could be done to the list of known

relationships, if, for example older historical trends are not of interest, or could be applied to potential targets, for example, ignoring adverts in a different language to the query item.

The next stage is to calculate similarity between the query item and other items in the same collection which are members of relationships of the type we are looking to infer, items not in such relationships are not of interest. We then calculate the semantic distance between the query and each of these, we refer to these items as "similar nodes" and call the semantic distance for each D1.

We then look at all items pointed to by the known relationships of each similar node, we refer to these collectively as "related nodes". If the number of similar nodes is large, we can choose to only follow relationships for a maximum number of similar nodes, preferring ones most similar to the query, in the results section we denote this parameter as L1. We then calculate the semantic distance between each related node and every other node in that space, which we call the "target nodes" and the distance D2. An item can be both a related node and a target node, but an item cannot be both the query and a target node. If the number of target nodes is large, we can limit the number of comparisons in the next stage by considering only a maximum number of targets for each related node, preferring the most similar, we denote this parameter as L2.

We discuss alternative scoring approaches in Sect. 6.3, but a simple scoring metric equivalent to the pre-selected items examples in the previous section is to determine the score for each target node by finding the largest value for $1-(D1+D2)/2$ (equivalent to Eq. 3) that creates a path to the target from the query item, where D1 is the semantic distance between the query and an item in the query's space (the similar node), which shares a relationship with an item in the target's space (the related node) which is of semantic distance D2 to the target node. This scoring system ranks items by the least combined-semantic-distance from a known relationship of the desired type.

For the discovery of similar nodes and target nodes, a clustering method could be used instead of our approach of selecting the L1 and L2 most similar items. In this paper we choose to use the nearest-neighbours approach instead of clustering to minimize the number of parameters and dataset specific tuning, although a clustering approach may assist in selecting more informative routes and potentially improve performance, however, this is left to future work.

4.3 Visualisation and provenance

In Fig. 5 we show a visualised example of several TSR routes for a query. The evaluation software can also produce interactive 3d plots which allow inspection of individual routes and the relevant nodes and labels, allowing some insight into the behaviour of the scoring algorithm.

Section 2.2 discussed the benefits of transparency and provenance for recommender systems. TSR makes use of a "black box" upstream embedding model to produce "white box" recommendations. While it is not possible to plainly describe why any two items are considered similar, the working of the algorithm in all later stages, such as items and known relationships considered, and the weighting of each, are fully

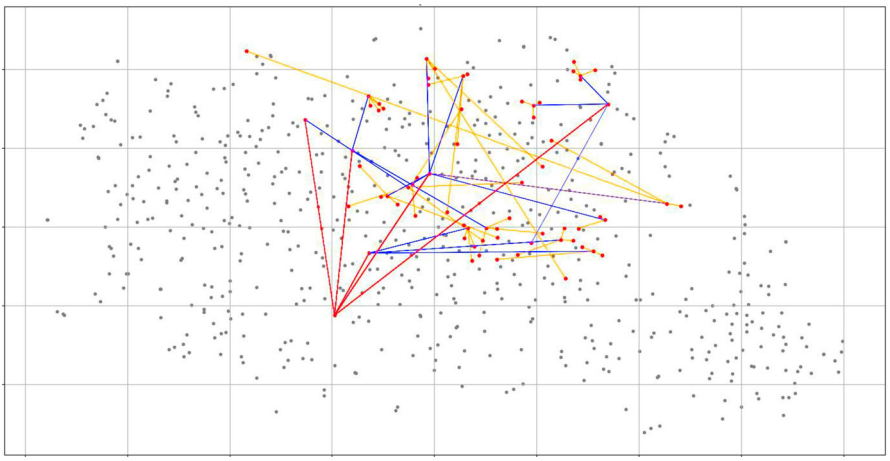


Fig. 5 A 2D t-SNE plot of IWSC item descriptions showing labelled and inferred relationships for a TSR query. Each route is comprised of three lines: $query\ node \rightarrow similar\ node$ (red), $similar\ node \rightarrow related\ node$ (blue), $related\ node \rightarrow target\ node$ (yellow). It can be seen that TSR uses a small number of known relationships (blue) to identify relevant areas of the search space to look for and rank targets including unlabelled items (colour figure online)

transparent. This is analogous to the justification a human decision maker might give, which for supply chain might typically cite existing relationships between similar companies that the expert considers relevant examples. In Sect. 7 we look in more detail at the provenance output for an example query.

5 Evaluation techniques

Various evaluation metrics are used in recommender system and information retrieval literature. As the IWSC dataset uses binary labels, and the total number of labels is small, we look at evaluation techniques which best reflect this.

Normalised Discounted Cumulative Gain (NDCG) [12] is a common evaluation metric in information retrieval literature. This is a graded relevance metric which rewards good results occurring sooner in the results list, however it does not penalise highly ranked negative items. As binary labels have no ideal order for positive items, we do not consider this a suitable metric.

Quantitative error metrics such as Root Mean Squared (RMS) error or Median Absolute Error are also common. Error metrics naturally favour scoring systems optimised to minimise loss such as learning-to-rank algorithms and require scores to fit the same range as the label values. For the IWSC dataset, as the labels are binary, the range is 0 to 1. However, scores output from TSR have no guarantee of symmetric distribution over the possible output range and are typically concentrated towards high-middle values due to averaging similarity scores making extreme values uncommon. Figure 6 shows the typical score distribution for the standard TSR algorithm TSR-a. In Sect. 6.3 we describe some alternative scoring algorithms with

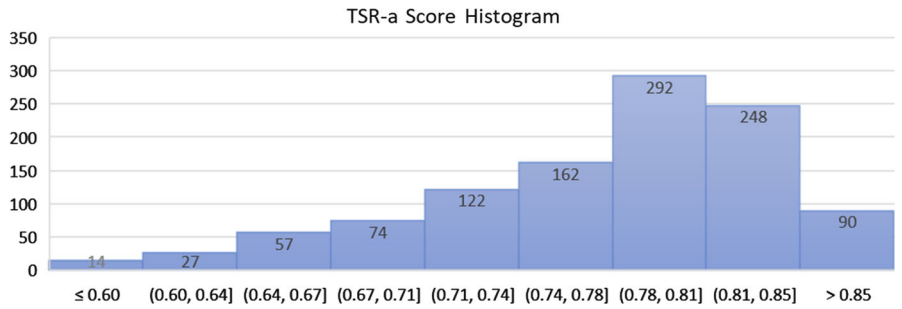


Fig. 6 Histogram of item scores produced by TSR-a

unbounded upper values. A scaling function can be applied after scores are calculated to fit them to a specific range, but this still does not guarantee the desired distribution and could be sensitive to outliers, such as unusually high scoring items, distorting error values.

For a binary labelled dataset, it is intuitive to set some threshold on the rankings and produce a confusion matrix, and take precision (P), recall (R), and f1 scores. As scores are not evenly distributed, there is no obvious score value to use as a threshold, so instead we look at some number of the top ranked items.

Due to the sparsity of labels in the dataset, the number and ratio of known positives and known negatives varies significantly between items and in many cases the number of known positives is smaller than typical values of K used for Precision at K. For this reason, we instead use R-Precision, setting the threshold at R, the number of true positives, and take the R most highly rated items to be predicted positive and all remaining to be predicted negative; at this threshold P, R, and f1 are equal. In the results section, we denote scores taken at this threshold as @R. A drawback of this approach is that we can only evaluate using known positives and known negatives, which is a minority of possible pairs in a sparse dataset. The difficulty of this evaluation task also varies with the ratio of known positives and negatives which is undesirable when evaluating datasets such as IWSC where the ratio varies greatly between items.

Finally, we look at techniques from the literature on implicit feedback. Techniques for implicit feedback have the desirable property of allowing us to expand the number of unique evaluation cases by enabling us to use unlabelled pairs of items (which for a sparse dataset is most possible item pairs) as implicit negative feedback. We use the common evaluation framework used by He et al. [9] and Koren [14], where we perform leave-one-out cross validation by, for each item, taking one known positive and 100 randomly selected other items (excluding known positives), and judging the ranking algorithm by ability to rank the known positive highly. The typical threshold used is that the known positive must be in the top 10 results, this Hit Ratio (HR) metric is denoted as HR@10. HR@5 refers to the known positive being in the top 5, and HR@1 as it being the highest rated item. We also show the mean and median values for the ranks of the known positives across all test cases.

It is of note that due to the random selection of negative items results may vary between runs. To ensure the results are representative we test each known posi-

tive against multiple random pools of implicit negatives. This significantly increases the compute time required for evaluation but minimises variation in scores between runs.

Having a fixed number of items in each evaluation and repeating with different random sets of items makes this metric well suited to datasets with uneven label distribution such as IWSC. We also consider the values to be quite intuitive as the random-algorithm performance for any HR@n is approximately n%, with ideal performance always being 100%. Mean and median positive label rank is in the range 0 to 100.

6 Results

We first use a neural language model to generate fixed length embeddings for all descriptions. In this study we use Universal Sentence Encoder (USE). This model was chosen as it shows good performance on a range of existing downstream tasks [6]. It is also of particular interest that this model was fine-tuned on the SNLI dataset [5], a set of sentence pairs labelled as contradiction, entailment, or unrelated; we speculate that this may require the model to learn similar linguistic features as are likely needed for the supply chain inference task as the ability to discern whether pairs of descriptions are entailed or contradictory is essential to human judgements for this task, in particular, in determining if companies serve similar supply chain roles. As the focus of this paper is in introducing TSR, we leave detailed investigation of the effects of upstream embedding models to future work.

6.1 Results for subset labelled tasks

Tables 2 and 3 show our results on the two IWSC-SL tasks introduced in Sect. 3. In these experiments we used the least-combined-cosine-distance scoring metric described in Sect. 4.2 and evaluate using metrics discussed in Sect. 5. All experiments are user-wise cold-start scenarios where the query item is treated as unseen, only the USE embedding of its description is known.

We set the parameters $L1 = 5$ and $L2 = 10$, for this scoring metric the value of these parameters has little impact on performance as only the best routes contribute to scoring, but it is observable that this inflates the mean positive rank as items lacking good routes are excluded from the results, which we treat as being given the worst possible rank. Using higher values for L1 and L2 produces more accurate values for

Table 2 Explicit feedback evaluation of TSR-a on the IWSC-SL tasks

Positive label name	Labelled items	Positive labels	Negative labels	F1 @R	RMS error	Median absolute error
SL_consumers	16	375	712	0.520	0.204	0.688
SL_suppliers	15	142	525	0.477	0.234	0.682

Table 3 Implicit feedback evaluation of TSR-a on the IWSC-SL tasks

Positive label name	Labelled items	Positive labels	HR @10	HR @5	HR @1	Median positive rank	Mean positive rank
SL_consumers	17	376	0.752	0.510	0.146	4	7.8
SL_suppliers	15	142	0.663	0.543	0.150	4	14.0

mean positive rank but significantly increases run time and does not alter the scores or order of high ranking items, hence the median positive rank is unaffected.

For the implicit feedback evaluations (HR and Positive Rank) we use one known positive, and a random pool of 100 not-known-positive items. We repeat this process 10 times for each positive label, using different random pools, and calculate the scores across all tests. Therefore, the number of test runs is always 10 times the number of positive labels. The number of labelled items and positive labels used in the implicit feedback tests is higher as we can additionally test items that lack any known negatives.

Our results show good performance on the IWSC-SL tasks, considering how few labels are available, achieving a hit-rate@10 of over 75%. It is notable that we see less than 9% worse performance on the SL_suppliers test despite having less than half the number of labels, showing that the algorithm can achieve good performance on subset-labelled tasks even when extremely few labels are available (142 labels in a dataset of 630 items). For both IWSC-SL tasks the frequency of the top ranked item being the known positive (when competing with 100 randomly selected others) HR@1 appears similar and is 14-15 times better than random.

6.2 Results for extra sparse labelling tasks

Tables 4 and 5 show our results on the two IWSC-ES tasks introduced in Sect. 3. The algorithm and parameters are the same as in the IWSC-SL tasks tests. The IWSC-ES tasks each have around half the number of positive labels as the IWSC-SL tasks, so a lower score should be expected.

In the IWSC-ES tasks we show significantly worse hit-rate, but smaller median absolute error and RMS error. We speculate that the lack of dense regions in the labels, due to the extreme sparsity and random distribution, makes identifying a particular known positive more difficult, but the better error values and F1 score indicate that the predicted scores are still effective for discerning good and bad results despite being less effective at a ranking a given good result highly.

Table 4 Explicit feedback evaluation of TSR-a on the IWSC-ES tasks

Positive label name	Labelled items	Positive labels	Negative labels	F1 @R	RMS error	Median absolute error
ES_consumers	39	115	198	0.549	0.167	0.560
ES_suppliers	46	90	259	0.350	0.177	0.572

Table 5 Implicit feedback evaluation of TSR-a on the IWSC-ES tasks

Positive label name	Labelled items	Positive labels	HR @10	HR @5	HR @1	Median positive rank	Mean positive rank
ES_consumers	51	207	0.221	0.119	0.018	36	43.0
ES_suppliers	48	92	0.197	0.129	0.055	32	47.7

6.3 Alternative scoring algorithms

The TSR-a scoring algorithm described previously, taking the score for a target as simply the minimum combined-semantic-distance (i.e. the semantic difference of the most similar known relationship to that query-target pair), is relatively simple to calculate and is both intuitive and easy to visualise (see Fig. 5). However, as only the shortest route to a target is considered, it does not factor in supporting evidence. For example, in the case of two targets with highly similar shortest distances from the query, if one had multiple high-quality routes and the other had only the one short route, we would intuitively be more confident to recommend the target with greater supporting evidence.

We test several variations of the scoring algorithm which boost the score when multiple good routes to the target are found. These approaches include boosting the score based on the number of routes (TSR b and c), taking the weighted sum of the scores for each route (TSR d, e, f, g, h, k, l, m, o, p, and q), and taking the sum of scores for each route but increasing the significance of distance (e.g. distance squared or cubed) (TSR i, j, and n). The results of these tests for the SL_consumers task are shown in Table 6 and a comprehensive comparison across all tasks is shown in Fig. 7. Detailed results of each scoring algorithm on each task can be found in the GitHub repository [20]. As these algorithms produce scores outside the range 0–1, we apply a simple scaling algorithm shown in Eq. (4).

$$f(s_i) = \frac{s_i - min(s)}{max(s) - min(s)} \tag{4}$$

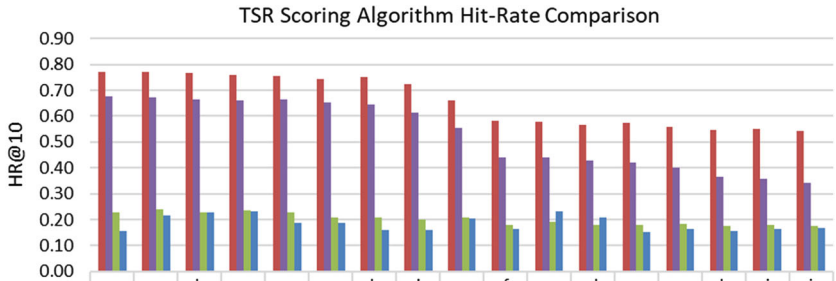
The scaling algorithms does not modify the order of results but ensures scores are within the same 0–1 range as the labels to make them suitable for error measurement. TSR-a produces scores in the range 0–1 without scaling, but we include a scaled version TSR-a* for comparison, as TSR-a rarely gives scores close to its bounds (see Fig. 6).

We find that most of these approaches perform either similarly to, or significantly worse than scoring by only the best route as in TSR-a. The scoring metrics that do perform better show slight improvement.

The best performing algorithm for the IWSC-SL tests is TSR-e, where we calculate the target score as the sum of score for the best route and half the score of the second-best route. This produced an improvement to HR@10 of 1.7% for the SL_consumers task and 1.2% for SL_suppliers but has the disadvantage of having a score distribution concentrated towards middle values, as extreme values would require either all routes

Table 6 Evaluation of alternative TSR algorithms on the IWSC SL_consumers task

Scoring algorithm	HR @10	HR @5	HR @1	Median positive rank	Mean positive rank	F1 @R	RMS error	Median absolute error
TSR-a	0.754	0.509	0.145	4	7.7	0.520	0.204	0.688
TSR-a*	0.754	0.509	0.145	4	7.7	0.520	0.195	0.481
TSR-b	0.548	0.364	0.115	8	11.5	0.541	0.120	0.319
TSR-c	0.573	0.385	0.133	7	10.9	0.544	0.120	0.309
TSR-d	0.565	0.373	0.124	7	11.1	0.544	0.122	0.322
TSR-e	0.771	0.532	0.163	4	7.6	0.530	0.204	0.584
TSR-f	0.582	0.408	0.158	7	10.5	0.549	0.146	0.456
TSR-g	0.742	0.536	0.185	4	7.8	0.533	0.192	0.523
TSR-h	0.767	0.538	0.152	4	7.5	0.531	0.196	0.508
TSR-i	0.543	0.362	0.112	8	11.5	0.541	0.121	0.320
TSR-j	0.550	0.359	0.117	8	11.6	0.541	0.120	0.318
TSR-k	0.750	0.538	0.179	4	7.9	0.525	0.207	0.605
TSR-l	0.723	0.529	0.189	4	8.1	0.536	0.189	0.525
TSR-m	0.771	0.530	0.151	4	7.5	0.523	0.170	0.433
TSR-n	0.577	0.385	0.135	7	10.7	0.541	0.121	0.320
TSR-o	0.659	0.466	0.181	5	9.2	0.539	0.143	0.452
TSR-p	0.758	0.533	0.158	4	7.5	0.531	0.165	0.456
TSR-q	0.558	0.372	0.119	8	11.2	0.541	0.120	0.325



	e	m	h	p	a	g	k	l	o	f	n	d	c	q	b	j	i
SL_consumers	0.77	0.77	0.77	0.76	0.75	0.74	0.75	0.72	0.66	0.58	0.58	0.57	0.57	0.56	0.55	0.55	0.54
SL_suppliers	0.68	0.67	0.67	0.66	0.66	0.66	0.65	0.61	0.55	0.44	0.44	0.43	0.42	0.40	0.37	0.36	0.34
ES_consumers	0.23	0.24	0.23	0.23	0.23	0.21	0.21	0.20	0.21	0.18	0.19	0.18	0.18	0.18	0.18	0.18	0.18
ES_suppliers	0.16	0.22	0.23	0.23	0.19	0.19	0.16	0.16	0.20	0.16	0.23	0.21	0.15	0.16	0.16	0.16	0.17

Fig. 7 Comparison of Hit rate of alternative TSR algorithms on all four IWSC tasks

to be very poor, or both routes to be very good, which is less common than only the best route being very good or bad. This may account for its comparatively high error values as error measurements will be high even for a correct ordering if values are concentrated towards the mid-range.

Another well performing algorithm is TSR-m, as given in Eq. (5), where r is the number of routes to the target, i is the rank of each route (where $i = 1$ is the route with the least combined-semantic-distance), and d_i is the combined-semantic-distance (Eq. 3) of route i. As the output scores S are not in the range 0–1, we apply a scaling function once all score values have been calculated. We omit the scaling function for clarity as it is already given in Eq. (4).

$$S = \sum_{i=1}^{r} \left(\frac{1}{d_i i^3} \right) \tag{5}$$

The algorithms TSR-o and TSR-p are the same as TSR-m except that the exponent of the route's rank, which the score is divided by, is 1 and 2 respectively; these variations perform significantly worse. It is interesting that when penalising the contribution of additional routes, we see sub-standard performance when the penalty is small, but above-standard performance when it is large. This would suggest that some ideal penalty function exists where additional routes do not overpower the score from the best route but still provide support in closely scored cases. It is possible that the best scoring penalty is a property of the distribution of the data and labels, and that the ideal penalty function may be dependent on the dataset. Testing of this property on other datasets, and alternative penalties for this dataset are left to future research.

7 Examples

To help illustrate the behaviour of TSR, we give some examples of output. Table 7 in the appendix shows the TSR-e results for the query company "Resmar Marine Safety"

using the SL_Consumers label set. The parameters used in this query are the same as in our empirical evaluation, and the query was treated as a user-wise cold start (only the description was used).

In this case, four out of the top five results are labelled potential consumers, and the other is unknown (not labelled). Inspecting the provenance output from TSR shows that, except for "Datum Electronics Limited", the recommendations were primarily based on Resmar Maine Safety's similarity to "Superyacht Doc" and "ProSafe Consultants Ltd", which the top four results are all labelled as potential consumers of.

Repeating this query using TSR-a produces a different set of top results. In this case we observe that the similarity between Resmar and Superyacht is the deciding factor for all of the top results. For further examples, we encourage the interested reader to refer to the following section on reproducibility.

8 Reproducibility

We have made available for download the full suite of evaluation tools and TSR implementation used in generating all results presented in this paper, along with the full experimental results and IWSC dataset, with and without the generated description embeddings used in these experiments. All these can be found in the repository [20].

In Sect. 6.3 we only describe in detail the best performing scoring algorithms, as many variations were tested. The full implementation of each can be found in the publicly available TSR implementation. Table 7 in the appendix shows an excerpt of one example TSR query, which was chosen for its clear demonstration of the method of operation of the scoring algorithms. The tools to inspect other queries of the operators choosing are provided in the public repository.

9 Conclusions

We have demonstrated the Transitive Semantic Relationships technique as an effective recommendation algorithm on datasets with very few labels and from cold starts. In particular we see good performance on the subset-labelling tasks of the Isle of Wight Supply Chain dataset also introduced in this paper. In or investigation of alternative item scoring methods we show that supporting evidence in the form of additional high-quality routes to a target can have a positive impact on performance, but that the weighting used can have a large impact on performance. Additionally, we find that the inclusion of additional routes in the scoring can have a negative effect if the labels are extremely sparse and not concentrated. Using TSR we set the baseline performance on the four recommendation tasks for the IWSC dataset. Our best performing algorithm TSR-e showing a hit-rate@10 of 77% and hit-rate@1 of 16% on the SL_consumers new user cold-start task.

The novel technique introduced in this paper provides an effective solution for the challenging problem of user-wise cold-starts in sparsely labelled and partially labelled

datasets, which are a known weakness of many existing recommender systems. The focus of this paper has been on introducing the TSR technique and IWSC dataset and tasks; both contributions open new avenues for further investigation into the properties of extremely sparse, and subset labelled datasets and additionally demonstrate the challenge and a potential solution to the user-wise cold-start problem. Future work may examine how TSR could be applied to expand the number of relationships in a partially labelled dataset to allow the use of algorithms that struggle with cold starts or require many training examples. The IWSC dataset also offers open challenges such as double cold starts (where no labels are given for either the user or target items), or prediction of the most likely relationship type (if any) for a given item pair.

Acknowledgements We thank Launch International LTD for their contribution of speculative supply chain labels for the IWSC dataset, under the direction of co-author Dr Li. This research is jointly funded by KnowNow Information LTD and the Engineering and Physical Sciences Research Council (EPSRC), project reference 1953880.

Appendix

See Table 7.

Table 7 Example results for an SL_Consumers query using TSR-e for the company "Resmar Marine Safety"

Name	Description text	Known relation	TSR-e rank
Resmar Marine Safety	Resmar specialise in boat safety equipment, fire safety apparatus, and industrial safety equipment. The boating safety equipment we supply includes Life Rafts, Life Jackets and Flotation aids	Query	–
Caversham Boat Services	Holiday Boat Hire—Narrowboats and Cruisers, Jetty services, Slipway, Engineering and Mooring	Consumer	1
Burgess Marine Ltd	Super yacht refit, 900 ton ship lift, steel and aluminium welding and fabrication, All aspects of commercial ship repair Support of WFSV, commercial ferry industry, Royal Navy Surface Fleet and commercial tonnage	Consumer	2
Green Marine Solutions	After completing three successful years on the Greater Gabbard wind farm the Marine Management team contracted by Fluor to plan, initiate and manage the Marine Coordination Centre have formed Green Marine Solutions Green Marine Solutions offer three packages to the Offshore Renewable industry:, (1) Marine Operations and Coordination. By packaging Marine Coordination, management and equipment procurement under one umbrella, GMS will work with clients to plan, run and continually develop their Marine Co-ordination centre and procedures. (SHORTENED DUE TO LENGTH)	Consumer	3
Motions Charters	Motion Charters is a family run business based in Hamble near Southampton. We offer a variety of luxury cruising boats, powerboats and race yachts which are all well maintained and fully equipped for your trip, whether you're enjoying a spot of gentle cruising or competing in a sailing event. We pride ourselves on friendly customer service and offer 24/7 support to ensure you have an enjoyable time on the water. Call us for more details and we'll find the best package to suit you and your guests	Consumer	4
Datum Electronics Limited	Datum Electronics is a world-leading supplier of marine shaft power meters. Our unique fully modular systems are capable of measuring the on-shaft torque and power of a ship on shafts from 150 to 1100 mm (and above) diameters. Shaft Power and Torsionmeters, systems suitable for ship trials or permanent installation into ships. (SHORTENED DUE TO LENGTH)	Unknown	5

The description text for each company is taken verbatim from the IWSC dataset [20], and were originally sourced from the websites of IWChamber [1], IWTechnology [2], and Marine Southeast [3]

References

1. IWChamber (2018). https://www.iwchamber.co.uk. Accessed 9 Oct 2018
2. IWTechnology (2018). http://iwtechnology.co.uk/. Accessed 9 Oct 2018
3. Marine Southeast (2018). http://www.marinesoutheast.co.uk/. Accessed 9 Oct 2018
4. Al-garadi MA, Varathan KD, Ravana SD, Ahmed E, Chang VI (2016) Identifying the influential spreaders in multilayer interactions of online social networks. J Intell Fuzzy Syst 31:2721–2735
5. Bowman SR, Angeli G, Potts C, Manning CD (2015) A large annotated corpus for learning natural language inference. In: Proceedings of the 2015 conference on empirical methods in natural language processing. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 632–642. https://doi.org/10.18653/v1/D15-1075
6. Cer D, Yang Y, Kong Sy, Hua N, Limtiaco N, St. John R, Constant N, Guajardo-Cespedes M, Yuan S, Tar C, Strope B, Kurzweil R (2018) Universal Sentence Encoder for English. In: Proceedings of the 2018 conference on empirical methods in natural language processing System demonstration. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 169–174. https://doi.org/10.18653/v1/D18-2029
7. Grady C, Lease M (2010) Crowdsourcing Document Relevance Assessment with Mechanical Turk. In: Proceedings of the NAACL HLT 2010 Workshop on creating speech and language data with Amazon's mechanical turk, June. Association for Computational Linguistics, Los Angeles, California, pp 172–179
8. Harper FM, Konstan JA (2015) The movielens datasets. ACM Trans Interact Intell Syst 5(4):1–19. https://doi.org/10.1145/2827872
9. He X, Liao L, Zhang H, Nie L, Hu X, Chua TS (2017) Neural Collaborative Filtering. In: Proceedings of the 26th International Conference on World Wide Web—WWW '17. ACM Press, New York, New York, USA, pp. 173–182. https://doi.org/10.1145/3038912.3052569
10. Herlocker JL, Konstan JA, Riedl J (2000) Explaining collaborative filtering recommendations. In: Proceedings of the 2000 ACM conference on computer supported cooperative work—CSCW '00. ACM Press, New York, New York, USA, pp 241–250. https://doi.org/10.1145/358916.358995
11. Hu M, Liu B (2004) Mining and summarizing customer reviews. In: Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining—KDD '04, p 168. https://doi.org/10.1145/1014052.1014073
12. Järvelin K, Kekäläinen J (2002) Cumulated gain-based evaluation of IR techniques. ACM Trans Inf Syst 20(4):422–446. https://doi.org/10.1145/582415.582418
13. Kong W, Li R, Luo J, Zhang A, Chang Y, Allan J (2015) Predicting search intent based on pre-search context. In: Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval—SIGIR '15, pp 503–512. https://doi.org/10.1145/2766462.2767757
14. Koren Y (2008) Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining. https://doi.org/10.1145/1401890.1401944
15. Koren Y, Bell R, Volinsky C (2009) Matrix factorization techniques for recommender systems. Comput (Long Beach Calif) 42(8):30–37. https://doi.org/10.1109/MC.2009.263
16. Li J, Chen X, Hovy E, Jurafsky D (2015) Visualizing and understanding neural models in NLP. https://doi.org/10.18653/v1/N16-1082
17. Maaten LVD, Hinton G (2008) Visualizing data using t-SNE. J Mach Learn Res 9:2579–2605
18. Musto C, Semeraro G, de Gemmis M, Lops P, Ferro N, Crestani F, Moens MF, Mothe J, Silvestri F, Di Nunzio GM, Hauff C, Silvello G (2016)) Learning word embeddings from wikipedia for content-based recommender systems. Springer, Cham, pp 729–734. https://doi.org/10.1007/978-3-319-30671-1_60
19. Pang B, Lee L (2005) Seeing stars. In: Proceedings of the 43th annual meeting of the association for computational linguistics—ACL '05 (1), pp 115–124. https://doi.org/10.3115/1219840.1219855
20. Ralph D, Li Y, Wills G, Green GN (2018) DavidRalph/TSR-Public. https://doi.org/10.5281/zenodo.3355448. https://github.com/DavidRalph/TSR-Public
21. See A, Liu PJ, Manning CD (2017) Get to the point: summarization with pointer-generator networks. In: Proceedings of the 55th annual meeting of the association for computational linguistics Volume 1 Long PAP. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 1073–1083. https://doi.org/10.18653/v1/P17-1099
22. Snow R, Connor BO, Jurafsky D, Ng AY, Labs D, St C (2008) Cheap and fast—but is it good? Evaluating non-expert annotations for natural language tasks. In: Proceedings of the conference on empirical

methods in natural language processing, EMNLP '08. Association for Computational Linguistics, Honolulu, Hawaii, pp 254–263
23. Suglia A, Greco C, Musto C, De Gemmis M, Lops P, Semeraro G (2017) A deep architecture for content-based recommendations exploiting recurrent neural networks. [UMAP2017]Proceedings 25th Conference on user modeling, adaptation, and personalization, pp 202–211. https://doi.org/10.1145/3079628.3079684
24. Vuurens JBP, Larson M, de Vries AP (2016) Exploring deep space: learning personalized ranking in a semantic space. In: Proceedings of the 1st workshop on deep learning on recommendation systems—DLRS 2016, pp 23–28. https://doi.org/10.1145/2988450.2988457
25. Xu Z, Chen C, Lukasiewicz T, Miao Y, Meng X (2016) Tag-aware personalized recommendation using a deep-semantic similarity model with negative sampling. https://doi.org/10.1145/2983323.2983874
26. Yosinski J, Clune J, Nguyen A, Fuchs T, Lipson H (2015) Understanding neural networks through deep visualization. arXiv:1506.06579
27. Yuan J, Shalaby W, Korayem M, Lin D, Aljadda K, Luo J (2016) Solving cold-start problem in large-scale recommendation engines: a deep learning approach. In: Proceedings of the 2016 IEEE International Conference on Big Data, Big Data 2016, pp 1901–1910. https://doi.org/10.1109/BigData.2016.7840810
28. Zhang F, Yuan NJ, Lian D, Xie X, Ma WY (2016) Collaborative knowledge base embedding for recommender systems. In: Proceedings of the 22nd ACM SIGKDD conference on knowledge discovery and data mining—KDD '16, pp 353–362. https://doi.org/10.1145/2939672.2939673
29. Zhang S, Yao L, Sun A, Tay Y (2019) Deep learning based recommender system: a survey and new perspectives. ACM Comput Surv 52(1):1–38. https://doi.org/10.1145/3285029
30. Zintgraf LM, Cohen TS, Adel T, Welling M (2017) Visualizing deep neural network decisions: prediction difference analysis, pp 1–12. arXiv:1702.04595