

Knowledge Graphs and Embeddings

an overview prepared for

Metrowest Boston Developers Machine Learning Group

Chris Winsor

6/16/2021

Agenda

- Overview
- Survey Paper

Cai, Zheng, Chang “A Comprehensive Survey of Graph Embedding: Problems, Techniques and Applications”, *Arxiv* (2017)

- Keras/Tensorflow Implementation

https://www.tensorflow.org/text/guide/word_embeddings

References:

- <https://towardsdatascience.com/an-introduction-to-knowledge-graphs-841bbc0e796e>
- Nickel, M., Murphy, K., Tresp, V., Gabrilovich, E. A Review of Relational Machine Learning for Knowledge Graphs. Proceedings of the IEEE (2015)
- <https://dl.acm.org/doi/abs/10.1145/2623330.2623623> (video + paper from 2014 – Kevin Murphy)
- https://web.stanford.edu/class/cs520/2020/notes/What_is_a_Knowledge_Graph.html
- Cai, H., Zheng, V. W. & Chang, K. C.-C. A Comprehensive Survey of Graph Embedding: Problems, Techniques and Applications. *Arxiv* (2017).
- Maas, A., Daly, R., Pham, P., Huang, D., Ng, A., Potts, C. “Learning Word Vectors for Sentiment Analysis” Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies

A Comprehensive Survey of Graph Embedding: Problems, Techniques and Applications

Hongyun Cai, Vincent W. Zheng, and Kevin Chen-Chuan Chang

Abstract—Graph is an important data representation which appears in a wide diversity of real-world scenarios. Effective graph analytics provides users a deeper understanding of what is behind the data, and thus can benefit a lot of useful applications such as node classification, node recommendation, link prediction, etc. However, most graph analytics methods suffer the high computation and space cost. Graph embedding is an effective yet efficient way to solve the graph analytics problem. It converts the graph data into a low dimensional space in which the graph structural information and graph properties are maximumly preserved. In this survey, we conduct a comprehensive review of the literature in graph embedding. We first introduce the formal definition of graph embedding as well as the related concepts. After that, we propose two taxonomies of graph embedding which correspond to what challenges exist in different graph embedding problem settings and how the existing work address these challenges in their solutions. Finally, we summarize the applications that graph embedding enables and suggest four promising future research directions in terms of computation efficiency, problem settings, techniques and application scenarios.

Index Terms—Graph embedding, graph analytics, graph embedding survey, network embedding

1 INTRODUCTION

GRAPHS naturally exist in a wide diversity of real-world scenarios, e.g., social graph/diffusion graph in social media networks, citation graph in research areas, user interest graph in electronic commerce area, knowledge graph etc. Analysing these graphs provides insights into how to make good use of the information hidden in graphs, and thus has received significant attention in the last few decades. Effective graph analytics can benefit a lot of applications, such as node classification [1], node clustering [2], node retrieval/recommendation [3], link prediction [4], etc. For example, by analysing the graph constructed based on user interactions in a social network (e.g., retweet/comment/follow in Twitter), we can classify users, detect communities, recommend friends, and predict whether an interaction will happen between two users.

Although graph analytics is practical and essential, most

can then be computed efficiently. There are different types of graphs (e.g., homogeneous graph, heterogeneous graph, attribute graph, etc), so the input of graph embedding varies in different scenarios. The output of graph embedding is a low-dimensional vector representing a part of the graph (or a whole graph). Fig. 1 shows a toy example of embedding a graph into a 2D space in different granularities. I.e., according to different needs, we may represent a node/edge/substructure/whole-graph as a low-dimensional vector. More details about different types of graph embedding input and output are provided in Sec. 3.

In the early 2000s, graph embedding algorithms were mainly designed to reduce the high dimensionality of the non-relational data by assuming the data lie in a low dimensional manifold. Given a set of non-relational high-dimensional data features, a similarity graph is constructed

A Review of Relational Machine Learning for Knowledge Graphs

Maximilian Nickel, Kevin Murphy, Volker Tresp, Evgeniy Gabrilovich

Abstract—Relational machine learning studies methods for the statistical analysis of relational, or graph-structured, data. In this paper, we provide a review of how such statistical models can be “trained” on large knowledge graphs, and then used to predict new facts about the world (which is equivalent to predicting new edges in the graph). In particular, we discuss two fundamentally different kinds of statistical relational models, both of which can scale to massive datasets. The first is based on latent feature models such as tensor factorization and multiway neural networks. The second is based on mining observable patterns in the graph. We also show how to combine these latent and observable models to get improved modeling power at decreased computational cost. Finally, we discuss how such statistical models of graphs can be combined with text-based information extraction methods for automatically constructing knowledge graphs from the Web. To this end, we also discuss Google’s Knowledge Vault project as an example of such combination.

Index Terms—Statistical Relational Learning, Knowledge Graphs, Knowledge Extraction, Latent Feature Models, Graph-based Models

I. INTRODUCTION

I am convinced that the crux of the problem of learning is recognizing relationships and being able to use them.

Christopher Strachey in a Letter to Alan Turing, 1954

form of relationships between entities. Recently, a large number of knowledge graphs have been created, including YAGO [4], DBpedia [5], NELL [6], Freebase [7], and the Google Knowledge Graph [8]. As we discuss in Section II, these graphs contain millions of nodes and billions of edges. This causes us to focus on *scalable* SRL techniques, which take time that is (at most) linear in the size of the graph.

We can apply SRL methods to existing KGs to learn a model that can predict new facts (edges) given existing facts. We can then combine this approach with information extraction methods that extract “noisy” facts from the Web (see e.g., [9, 10]). For example, suppose an information extraction method returns a fact claiming that Barack Obama was born in Kenya, and suppose (for illustration purposes) that the true place of birth of Obama was not already stored in the knowledge graph. An SRL model can use related facts about Obama (such as his profession being US President) to infer that this new fact is unlikely to be true and should be discarded. This provides us a way to “grow” a KG automatically, as we explain in more detail in Section IX.

The remainder of this paper is structured as follows. In Section II we introduce knowledge graphs and some of their properties. Section III discusses SRL and how it can be applied

Knowledge Graph

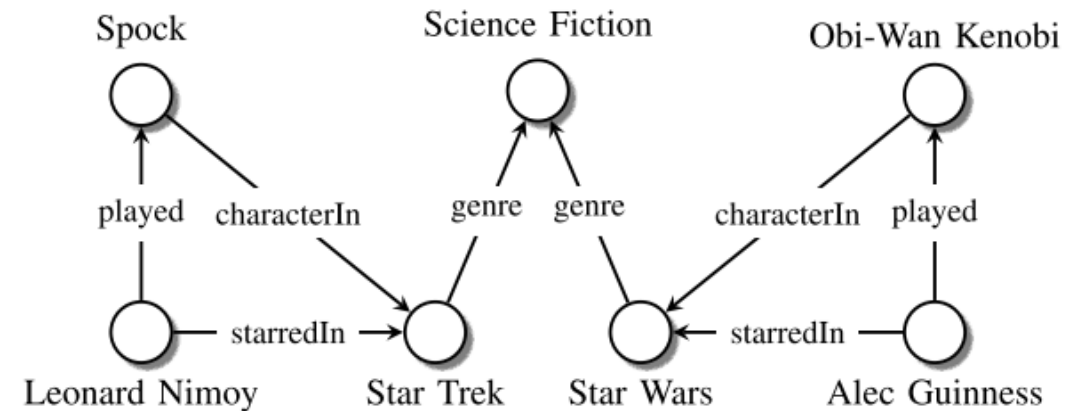
- A “graph structured knowledge base ... that stores factual information in (the) form of relationships between entities”.
- Typically scraped from public internet sources (natural text, images)
- Uses the [Resource Description Framework](#) (RDF) to represent facts as Subject-Predicate-Object (SPO) “triples”

Example

Leonard Nimoy was an actor who played the character Spock in the science-fiction movie Star Trek

SPO triples:

<i>subject</i>	<i>predicate</i>	<i>object</i>
<i>(LeonardNimoy,</i>	<i>profession,</i>	<i>Actor)</i>
<i>(LeonardNimoy,</i>	<i>starredIn,</i>	<i>StarTrek)</i>
<i>(LeonardNimoy,</i>	<i>played,</i>	<i>Spock)</i>
<i>(Spock,</i>	<i>characterIn,</i>	<i>StarTrek)</i>
<i>(StarTrek,</i>	<i>genre,</i>	<i>ScienceFiction)</i>



Why Knowledge Graph?

- Highly scalable (can handle large quantities)
- Ability to handle unstructured data
- To establish a base for subsequent ML-based learning, insights, predictions
- Example:
 - Troll Facebook movie reviews (unstructured text) into a standard format
 - Use that repository for movie recommendations, predict future academy awards, revenue prediction, etc.

KG is front end of (very large) data pipeline

Examples

Google	Knowledge Graph Acquired Freebase in 2010
Facebook	Graph API to Facebook Social Graph https://developers.facebook.com/docs/graph-api/
YAGO	Open source at Max Planck Institute for Computer Science in Saarbrücken Extracted from Wikipedia and other sources
DBpedia	https://www.dbpedia.org/ Maintained at University of Mannheim and Leipzig University Extracted from (and references to) Wikipedia.
Microsoft	(Teach-to-fish, give you fish) Graph API https://docs.microsoft.com/en-us/graph/overview Academic Knowledge Graph (MAKG) https://makg.org/ Knowledge Mining API (to Satori, Bing QnA, Enterprise Dictionary) https://www.microsoft.com/en-us/research/project/knowledge-mining-api/
Wikipedia	https://www.wikidata.org/wiki/Wikidata:Main_Page An open data source (welcomes responsible bots) and mined by all above WikiData is Wikipedia's own KG
IBM	https://researcher.watson.ibm.com/researcher/view_group.php?id=7140
CMU NELL	Never Ending Language Learning http://rtw.ml.cmu.edu/rtw/kbbrowser/

In summary: Very active and early on. Positions being staked out. Implementations are still research-ey

DBpedia

- Extracted from Wikipedia
- 2016-04 release = 6.0 million entities: 1.5M persons, 810k places, 135k music albums, 106k films, 20k video games, 275k organizations, 301k species and 5k diseases
- 9.5 billion RDF triples (1.3 billion from English edition Wikipedia and 5.0 billion from other language editions).

New Tab x Amazon x Amazon x Inbox (17 x Knowledge x nell know x ML Read the x what is g x What is C x grasshop x +

google.com/search?q=grasshopper&sxsrf=ALeKk01qTe_aHYwca4Rsg3lwUTIsIIIPnw%3A1623587663605&ei=T_vFYKKzJMG3tQbmm6aoDw&oq=grassh...

Apps Bug 3617: PreSens... Will Koehrsen (@ko... Home Page - COM... Tingjian Ge SQ-Dist SW-Releases SW-training SQA 3.1.3 Other bookmarks Reading list

Google

grasshopper

All Images News Videos Shopping More Settings Tools

About 32,900,000 results (0.86 seconds)

Ad · www.grasshopper.com/

Grasshopper™ Official Site - Get Your Business Phone Number

Look Like A National Company Or Establish A More Local Presence. Know it's a Business Call. Use your existing phone. Custom greeting. Vanity Numbers. Multiple extensions. Multiple plan options. Toll free & local numbers. Wifi Calling. No hardware required.

Pricing & Plans
Affordable plans for your business.
Billed Monthly or Annually.

Free 7-Day Trial
No Credit Card. No Commitment.
Try Grasshopper, On Us.

How It Works
Tons of great features
Use Your Existing Phone

Get A Toll Free Number
Customized vanity numbers
800 and 888 Numbers

https://grasshopper.com

Grasshopper Virtual Phone System | Manage Your Calls Online

Grasshopper provides you with a second phone number. **Grasshopper** is a full virtual phone system (calls, texts, custom greetings, extensions, inbound fax, and ...

Contact Us · How It Works · Sign Up · Desktop + Mobile Apps


See results about


- Grasshopper Company
- Grasshopper Insects
- Grasshopper 3D Programming language
- Grasshopper Drink
- The Grasshopper Company The Grasshopper Company is headquartered in Moundridge, ...


New Tab × Amazon Wo × Amazon Wo × Inbox (17,01 × Knowledge × nail project × Knowledge × FX 6 Steps to C × +

← → ↻ 🏠 🔒 webfx.com/internet-marketing/how-to-get-your-site-into-googles-knowledge-graph.html 🔍 ☆ 📄 📄 ⚙️ 🎵 👤 ⋮

📱 Apps 🛒 Bug 3617: PreSens... 🐦 Will Koehrsen (@ko... 📺 Home Page – COM... 🔄 Tingjian Ge 📄 SQ-Dist 📄 SW-Releases 📄 SW-training 📄 SQA » 📁 Other bookmarks 📖 Reading list

 REVENUE DRIVEN FOR OUR CLIENTS
\$2,416,945,839 ⓘ

 CLIENT LOGIN

 **888-256-9448**

WebFX Digital Marketing That Drives Results®

SEO & Lead Generation ▾ Ecommerce ▾ UX & Interactive ▾ Our Technology ▾ Who We Are ▾

Get a proposal 🚀

HOME / INTERNET MARKETING /

6 Steps to Getting in Google's Knowledge Graph

Google's Knowledge Graph is one of the most advanced features of its search engine.

View our Digital Services

2014

Name	# Entity types	# Entity instances	# Relation types	# Confident facts (relation instances)
<i>Knowledge Vault (KV)</i>	1100	45M	4469	302M
DeepDive [32]	4	2.7M	34	7M ^a
NELL [8]	271	5.19M	306	0.435M ^b
PROSPERA [30]	11	N/A	14	○ 0.1M
YAGO2 [19]	350,000	9.8M	100	4M ^c
Freebase [4]	1,500	40M	35,000	○ 637M ^d
Knowledge Graph (KG)	1,500	570M	35,000	18,000M ^e

Current Research Directions / Activity

- Dynamic Knowledge Graphs (McKallum <https://arxiv.org/abs/1810.05682>)
- Adding/changing entities or predicates (semantic structure) to existing KG (no re-training). Similar to “Dynamic KG” above.
- “Graph stream processing”

Theory

Historical Context

- Currently:
 - A means to aggregate and store knowledge scraped from internet
- Historically:
 - W3C Semantic Web - intended as a standardized machine-friendly way to express meaning and relationships in websites and web data.
 - Schema.org (consortium establishing standardized ontology)
- Webmasters did not code W3C so...
 - Google (et al) repurposed for use in KG

Knowledge Graphs leverage RDF (SPO, Schema Vocabulary, SPARQL) from W3C and (frequently) schema from schema.org

W3C RDF and SPO

- SPO (Subject-predicate-object) A triple that defines a “fact”.
 - Subject and Object are Entities, Predicate is relationship.
 - RDF (Resource Description Framework) defines a directed labeled graph – a set of SPOs
- RDF specification includes:
 - the [SPARQL Query Language](#) [[SPARQL11-OVERVIEW](#)];
 - the [RDF Schema vocabulary](#) [[RDF11-SCHEMA](#)].
- A class hierarchy
 - <https://www.w3.org/TR/rdf-schema/>
 - <https://www.w3.org/TR/rdf-sparql-query/>

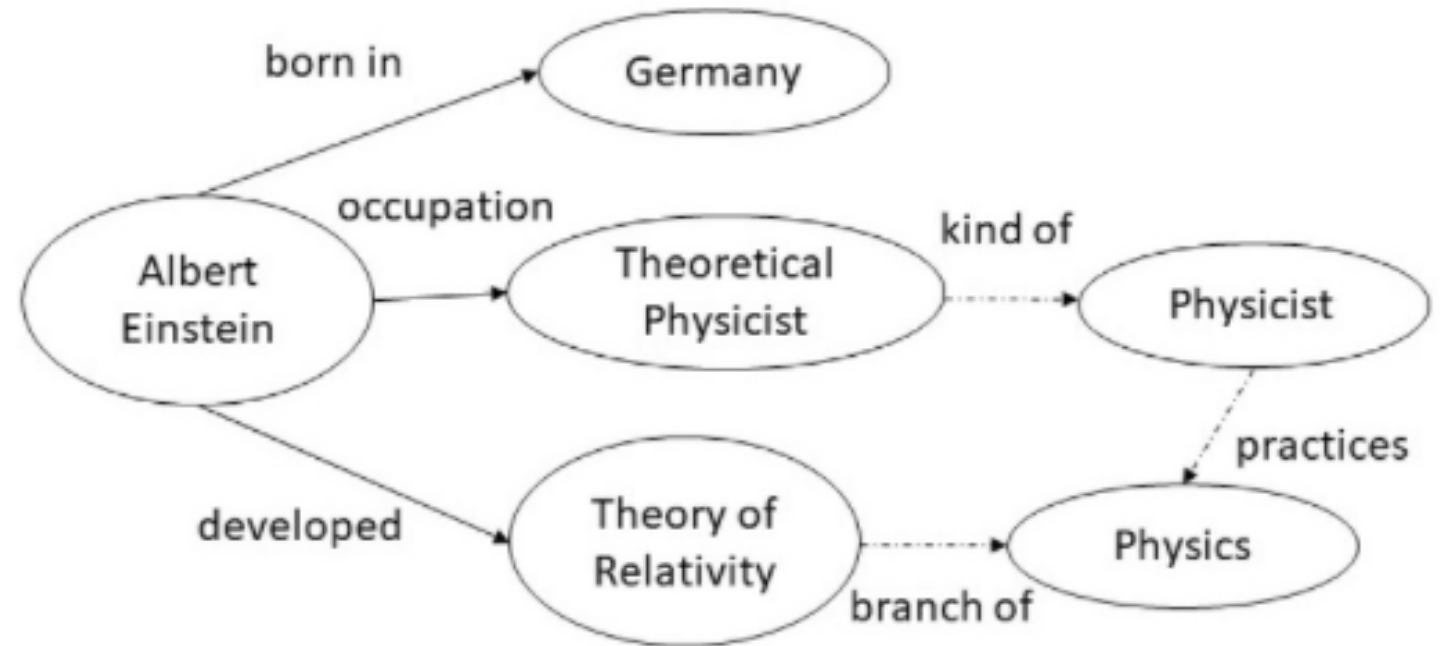
Fixed/Open Lexicon

- Fixed Lexicon (schema-based)
- Open Lexicon (schema free)
 - i.e. OpenIE (Stanford University) = NLP to SPO using “surface names”

Example

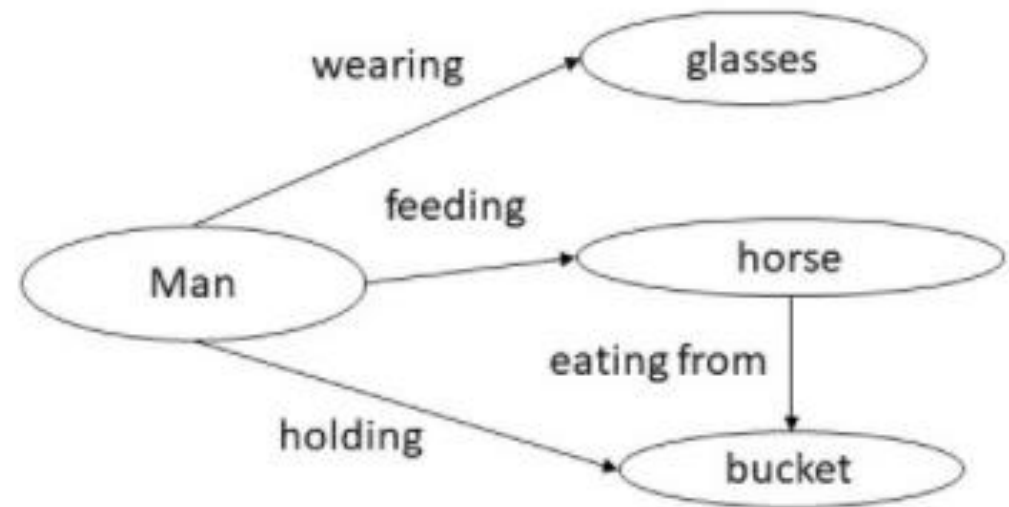
- Entity and relation extraction from natural text.

Albert Einstein was a **German-born theoretical physicist** who developed the **theory of relativity**.



Example

- Entities from object detector: (man, glasses, horse, bucket)
- Relationships: (wearing, feeding, holding, eating from)



Quality and Means of Construction

Method	Schema	Examples
Curated	Yes	Cyc/OpenCyc [23], WordNet [24], UMLS [25]
Collaborative	Yes	Wikidata [26], Freebase [7]
Auto. Semi-Structured	Yes	YAGO [4, 27], DBPedia [5], Freebase [7]
Auto. Unstructured	Yes	Knowledge Vault [28], NELL [6], PATTY [29], PROSPERA [30], DeepDive/Elementary [31]
Auto. Unstructured	No	ReVerb [32], OLLIE [33], PRISMATIC [34]

- Curated (manually built) and Collaborative (infoboxes) give highest quality
- Unstructured text, automated classification give lower confidence facts

The Problem

- KG Datasets are huge
 - Millions of entities, Billions of relations
 - Any entity can be related to any other
 - E.g. 1M people so $(1 * 10^6)^2$ possible “friend” relations
 - But a person typically has <100 friends (i.e. graph is sparse)
- Duplicate/similar semantics using different ground expression:
 - “Develop”, “create”, “invent” (semantically close)
 - “Obama”, “President Biden”, “the 46th President of the U.S.”
- Different semantics using same ground representation
 - Guinness (the beer), Guinness (the actor)

In summary: KG are a (large) pool of data - difficult to use directly

SRL (Statistical Relational Learning)

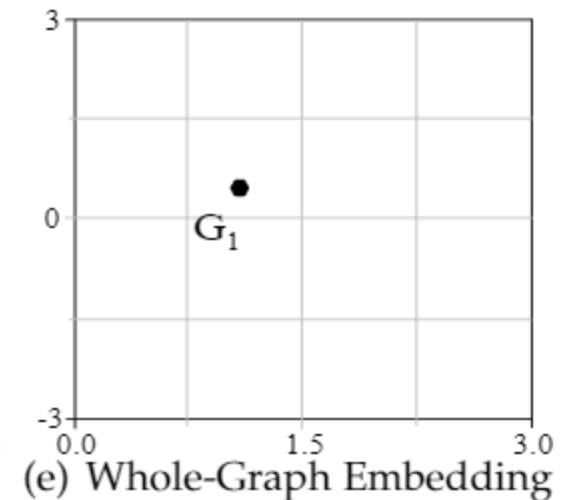
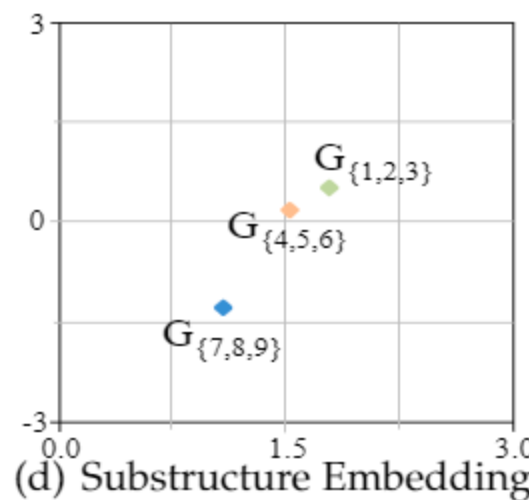
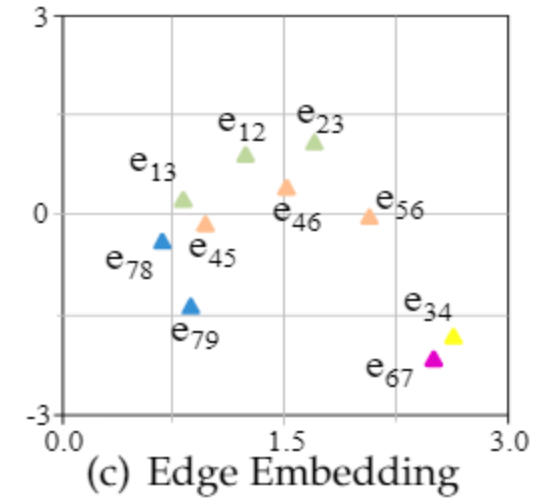
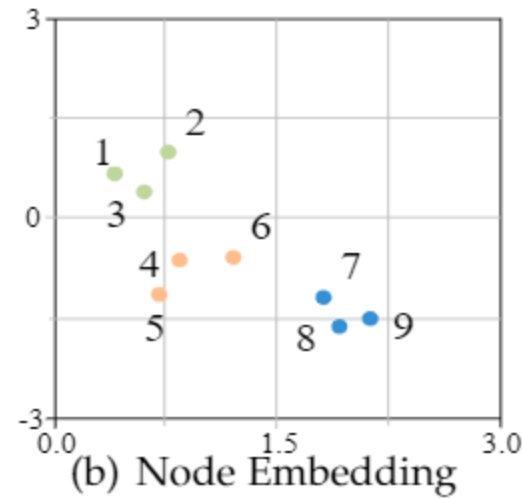
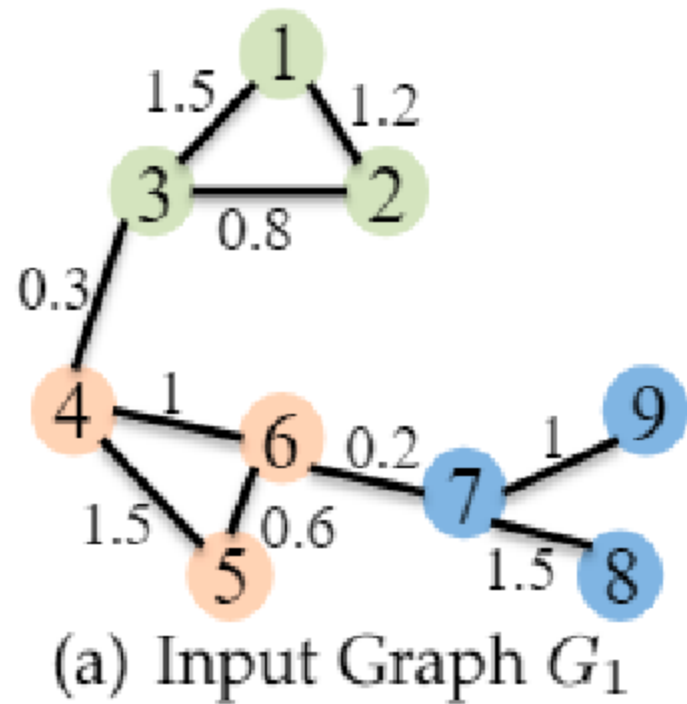
“We ... assume that all the entities and (types of) relations in a knowledge graph are known. However, triples are assumed to be incomplete and noisy; entities and relation types may contain duplicates”.

My read: Like any Machine Learning method, the goal is to parameterize what we can glean from the data (the KG) and capture that as a Model. SRL is the process. Embeddings are the embodiment of our knowledge.

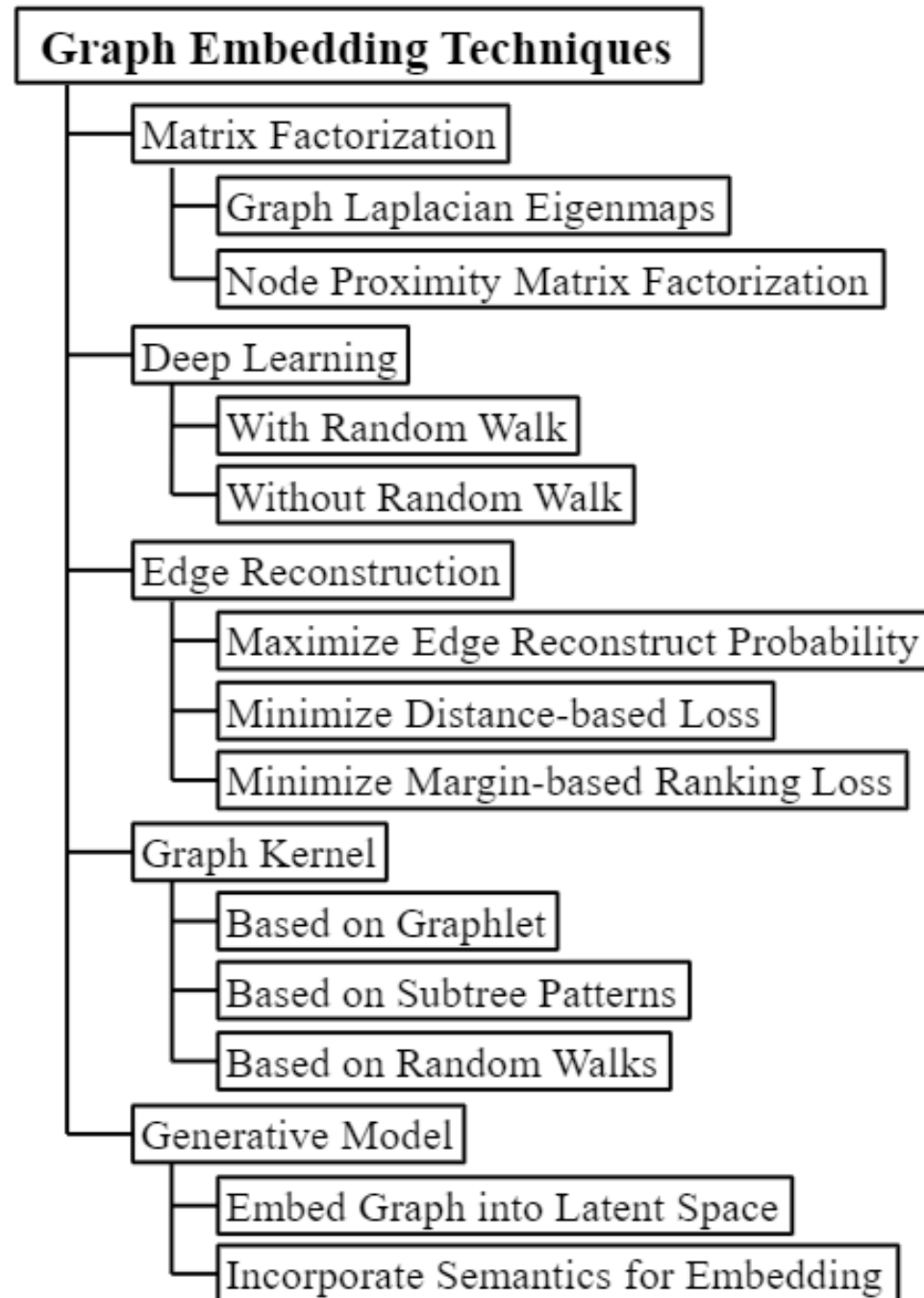
Graph Embedding Output

- A set of low dimensional vectors representing (part of) a graph.
 - Node Embedding
 - Node-pair (Edge)
 - Subgraph
 - Whole graph
- Driven by application

Graph Embedding Output



Techniques



Model Types

- Latent Feature: explains triples through latent variables (triples are independent given a latent feature)
 - RESCAL
 - Multi-layer Perceptron
- Latent Distance: relationship is based on distance
 - Structured embedding
- Graph Feature: explains triples through features that are present
 - Path Ranking

much more here...

Embeddings...

Embedding maps low-level feature (word or object) to higher-level concept.

- a) Neural networks operate on numeric, not nominal, values.
- b) The number of embedding values is driven by need for detail (application dependent).

I like knowledge graphs.

I like databases.

I enjoy running.

I => 21034

like => 44033

enjoy => 44033

knowledge graphs => 554433

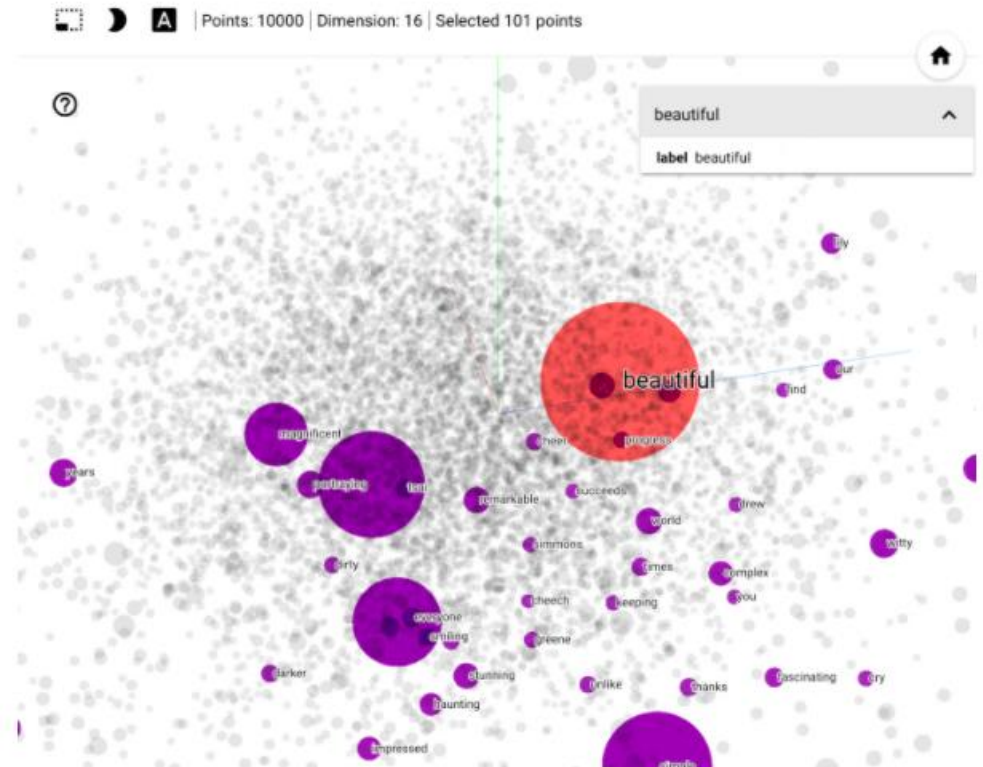
databases => 554433

Next...

TensorFlow Implementation

https://www.tensorflow.org/text/guide/word_embeddings

- Introduction to word embeddings
- Train Keras model for sentiment classification
- Visualize using Embedding Projector



Requirements and Steps

Equipment:

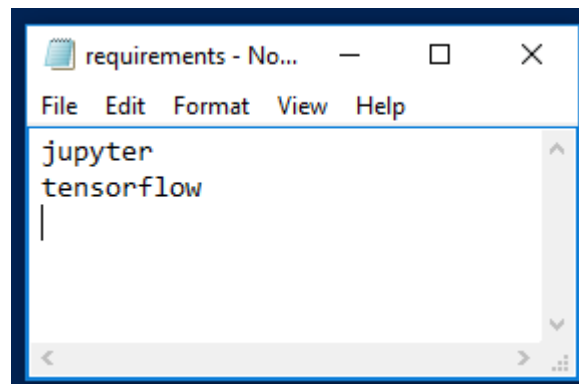
- Python with pip (`$python -m pip --version`)
- TensorFlow
- Jupyter Notebook

Procedure:

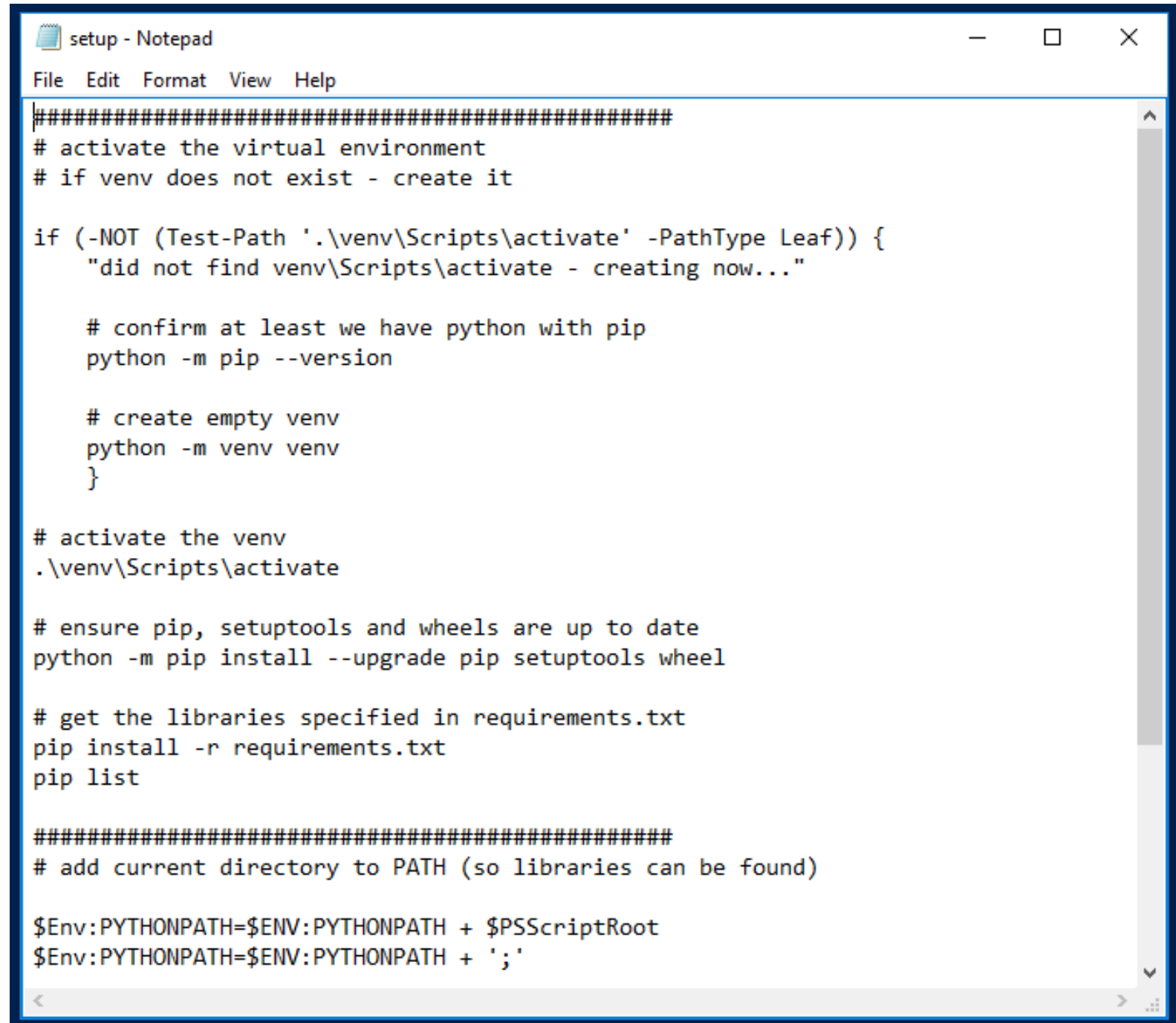
- Download notebook
https://storage.googleapis.com/tensorflow_docs/text/docs/guide/word_embeddings.ipynb
- Modules (venv is recommended - see next slide) or...
 - `python -m pip install tensorflow`
 - `Python -m pip install Jupyter`
- The notebook will download database, create embeddings

venv

- setup.ps1
- requirements.txt



```
File Edit Format View Help
jupyter
tensorflow
|
```



```
File Edit Format View Help
#####
# activate the virtual environment
# if venv does not exist - create it

if (-NOT (Test-Path '.\venv\Scripts\activate' -PathType Leaf)) {
    "did not find venv\Scripts\activate - creating now..."

    # confirm at least we have python with pip
    python -m pip --version

    # create empty venv
    python -m venv venv
}

# activate the venv
.\venv\Scripts\activate

# ensure pip, setuptools and wheels are up to date
python -m pip install --upgrade pip setuptools wheel

# get the libraries specified in requirements.txt
pip install -r requirements.txt
pip list

#####
# add current directory to PATH (so libraries can be found)

$Env:PYTHONPATH=$Env:PYTHONPATH + $PSScriptRoot
$Env:PYTHONPATH=$Env:PYTHONPATH + ';'

<
```


Large Movie Review (Dataset + Paper/Model)

<http://ai.stanford.edu/~amaas/data/sentiment/>

- Lexical meaning vs Sentiment
 - Meaning of the word vs feeling of the reader

Learning Word Vectors for Sentiment Analysis

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang,

Andrew Y. Ng, and Christopher Potts

Stanford University

Stanford, CA 94305

[amaas, rdaly, ptpham, yuze, ang, cgpotts]@stanford.edu

Abstract

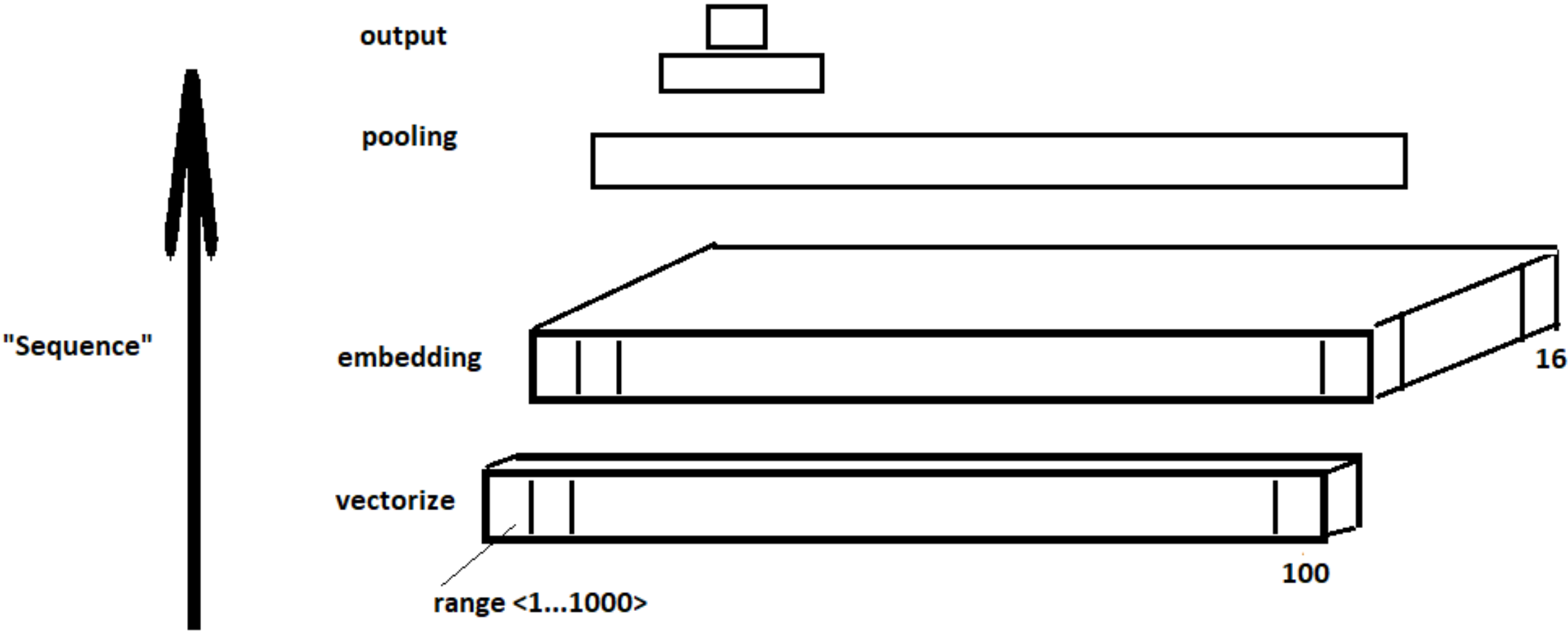
Unsupervised vector-based approaches to semantics can model rich lexical meanings, but they largely fail to capture sentiment information that is central to many word meanings and important for a wide range of NLP tasks. We present a model that uses a mix of unsupervised and supervised techniques to learn word vectors capturing semantic term-document information as well as rich sentiment content. The proposed model can leverage both continuous and multi-dimensional sentiment information as well as non-sentiment annotations. We instantiate the model to utilize the document-level sentiment polarity annotations present in many online documents (e.g. star ratings). We evaluate the model using small, widely used sentiment and subjectivity corpora and find it out-performs several previously introduced methods for sentiment classification. We also introduce a large dataset of movie reviews to serve as a more robust

recognition, part of speech tagging, and document retrieval (Turney and Pantel, 2010; Collobert and Weston, 2008; Turian et al., 2010).

In this paper, we present a model to capture both semantic and sentiment similarities among words. The semantic component of our model learns word vectors via an unsupervised probabilistic model of documents. However, in keeping with linguistic and cognitive research arguing that expressive content and descriptive semantic content are distinct (Kaplan, 1999; Jay, 2000; Potts, 2007), we find that this basic model misses crucial sentiment information. For example, while it learns that *wonderful* and *amazing* are semantically close, it doesn't capture the fact that these are both very strong positive sentiment words, at the opposite end of the spectrum from *terrible* and *awful*.

Thus, we extend the model with a supervised sentiment component that is capable of embracing many social and attitudinal aspects of meaning (Wil-

Tensorflow example



- Go to the notebook
- Review the database (files)
 - /pos /net
 - Readme
- Walk the Jupyter notebook

Thank You

Notation and Definitions

Graph	$G = (V, E)$
Knowledge graph	A directed graph where nodes are entities and edges are subject-property triple facts.
Triplet	$\langle h, r, t \rangle$ where $h, t \in V$ and $r \in E$
Proximity measure	The graph property to be preserved in the embedded space
First order proximity	Weight of edge from one node to another (weight of the relation)
Embedding	Given the input of a graph $G = (V, E)$ and dimensionality of embedding d where ($d \ll V $), embedding converts G into a d -dimensional space in which the graph property is preserved as much as possible.

Factor Graph

- A **factor graph** is a type of probabilistic graphical model. A factor graph has two types of nodes:
 - Random variable
 - Factor: used to evaluate relations between variables.

Google Knowledge Graph vs Google Knowledge Vault

- Knowledge Graph = product
 - Basis of google search
- Knowledge Value = research.
 - focusing on certainty. Each fact associated with confidence score and provenance
 - Seen as a precision/recall problem