Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Τεχνολογία Πληροφορικής & Υπολογιστών

Εκμάθηση σχέσεων από γράφους γνώσης με τεχνολογίες μηχανικής μάθησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΚΑΤΣΟΥΛΗ ΝΙΚΟΛΕΤΤΑ

Επιβλέπων : Γεώργιος Στάμου
Αν. Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2019

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Τεχνολογία Πληροφορικής & Υπολογιστών

Εκμάθηση σχέσεων από γράφους γνώσης με τεχνολογίες μηχανικής μάθησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΚΑΤΣΟΥΛΗ ΝΙΚΟΛΕΤΤΑ

Επιβλέπων : Γεώργιος Στάμου
Αν. Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 12η Ιουλίου 2019.

.....	.....	.....
Γεώργιος Στάμου Αν. Καθηγητής Ε.Μ.Π.	Ανδρέας-Γεώργιος Σταφυλοπάτης Καθηγητής Ε.Μ.Π.	Νικόλαος Παπασπύρου Αν. Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2019

Κατσούλη Νικολέττα

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Οι γράφοι γνώσης περιέχουν σημαντικές πληροφορίες για τον κόσμο με τη μορφή οντοτήτων και σχέσεων μεταξύ αυτών. Ωστόσο, όχι μόνο είναι δύσκολο να χρησιμοποιηθούν σε εφαρμογές Μηχανικής Μάθησης, αλλά έχουν και ελάττωμα σχετικά με την πληρότητα και την ορθότητα των δεδομένων τους. Το Link Prediction ασχολείται με την πρόβλεψη της ύπαρξης ή της πιθανότητας ορθότητας των πληροφοριών σε έναν Γράφο Γνώσης, καθώς το Statistical Relational Learning επικεντρώνεται στη δημιουργία στατιστικών μοντέλων για σχεσιακά δεδομένα. Τα Translational Distance Models είναι μια μέθοδος που χρησιμοποιείται ευρέως για την αντιμετώπιση αυτών των προβλημάτων, η οποία χρησιμοποιεί βασίζεται σε scoring functions με κριτήριο την απόσταση και υπολογίζει την αξιοπιστία ενός γεγονότος ως την απόσταση μεταξύ των embeddings των οντοτήτων.

Το TransE είναι ένα απλό και αποδοτικό μοντέλο, το οποίο λαμβάνει υπόψη μόνο τις άμεσες σχέσεις μεταξύ οντοτήτων. Από την άλλη πλευρά, το Path-based TransE (PTransE) δημιουργεί, επιπλέον, μονοπάτια σχέσεων πολλαπλών βημάτων στον Γράφο Γνώσης, αλλά υστερεί στην χρονική πολυπλοκότητα. Σε αυτή τη διπλωματική εργασία, προτείνουμε ένα συνδυασμό αυτών των δύο μοντέλων. Στόχος μας είναι να διατηρήσουμε την αποτελεσματικότητα του TransE και να επωφεληθούμε από τις πρόσθετες γνώσεις του PTransE. Ως εκ τούτου, προτού προχωρήσουμε στην εκπαίδευση, προσθέτουμε τις συνθέσεις σχέσεων και τις αντίστροφες σχέσεις στον Γράφο Γνώσης, υπολογίζουμε την αξιοπιστία κάθε διαδρομής και τροποποιούμε κατάλληλα το margin-based loss function του TransE.

Αξιολογούμε την προτεινόμενη μέθοδο σε δύο σύνολα δεδομένων, FB15K και SNOMED CT. Το δευτέρο βασίζεται σε αξιώματα Web Ontology Language που παρέχουν έναν τυπικό λογικό ορισμό των εννοιών. Η αξιολόγηση στο FB15K δεν επιτυγχάνει ιδιαίτερα καλά αποτελέσματα, αλλά στο SNOMED CT το μοντέλο μας παρουσιάζει σημαντική και συνεπή βελτίωση.

Λέξεις κλειδιά

Γράφος Γνώσης, Link Prediction, Στατιστική Σχεσιακή Μάθηση, Μηχανική Μάθηση, Translational Distance Models

Abstract

Knowledge Graphs contain rich information about the world in the form of entities and relationships between them. However, not only it is difficult to take advantage of them in Machine Learning tasks, but also they have flaws in correctness and completeness. Therefore, Link Prediction is concerned with predicting the existence or probability of correctness of information in a Knowledge Graph, while Statistical Relational Learning is focused on the creation of statistical models for relational data. Translational Distance Models are a widely used method to tackle these problems, which exploits distance-based scoring functions and measure the plausibility of a fact as the distance between the entities' embeddings, usually after a translation carried out by the relation's one.

TransE is a simple and efficient model, which takes into consideration only direct relations between entities. On the other hand, Path-based TransE (PTransE) builds also multiple-step relation paths on Knowledge Graph, but it has disadvantages in time complexity. In this diploma thesis, we propose a combination of these two models. We aim to maintain the efficiency of TransE and take advantage of the PTransE's extra knowledge. Hence, before training, we add relation paths and reverse relations to the Knowledge Graph, calculate the reliability of each path and modify appropriately the TransE's margin-based loss function.

We evaluate the proposed method on two data sets, FB15K and SNOMED CT. The second one contains Web Ontology Language axioms that provide a formal logical definition of the concept. Evaluation on FB15K do not achieve particularly better results, but on SNOMED CT our model provides significant and consistent improvement.

Key words

Knowledge Graph, Link Prediction, Statistical Relational Learning, Machine Learning, Translational Distance Models

Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω τον επιβλέποντά μου Γιώργο Στάμου για την καθοδήγηση, αλλά και την ελευθερία που μου προσέφερε στα πλαίσια της διπλωματικής εργασίας. Επιπλέον, οφείλω να ευχαριστήσω τον Αλέξη Μανδαλιό για την πολύτιμη βοήθεια, τις ιδέες και την υπομονή του.

Ακόμη, ευχαριστώ την οικογένειά μου για τη στήριξή τους καθόλη τη διάρκεια των σπουδών μου και του φίλους μου (Μαρίλη, Αθηνά, Κωνσταντίνο, Μαριτίνα και Νίκο) για την κατανόηση και υποστήριξη. Τέλος, θα ήθελα να ευχαριστήσω τον Δομήνικο για τη συμπαράσταση και τις συμβουλές που μου παρείχε.

Κατσούλη Νικολέττα,

Αθήνα, 12η Ιουλίου 2019

Contents

List of Tables

List of Figures

List of Algorithms

I

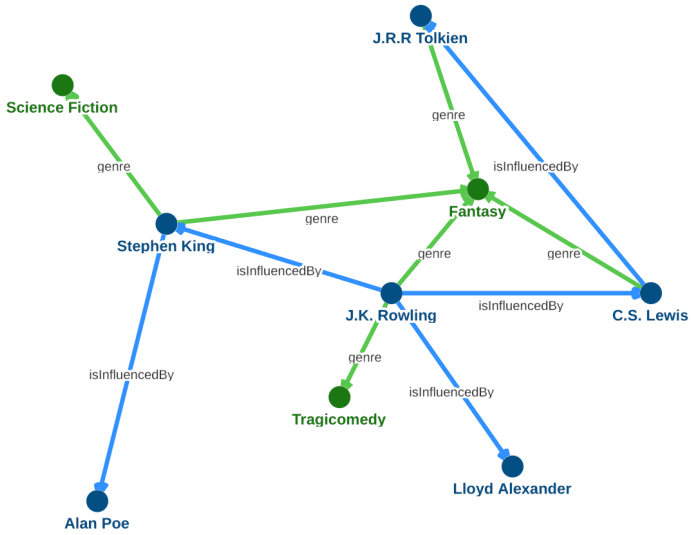
Κείμενο στα Ελληνικά

I.1 Εισαγωγή

I.1.1 Κίνητρο

Οι Γράφοι Γνώσης (KGs) παρέχουν δομημένη πληροφορία με τη μορφή οντοτήτων και σχέσεων μεταξύ τους. Οι κόμβοι αντιπροσωπεύουν οντότητες, ενώ οι ακμές αντιπροσωπεύουν υπάρχουσες σχέσεις. Οι KGs είναι ένα χρήσιμο εργαλείο για την αναπαράσταση της γνώσης και χρησιμοποιούνται όχι μόνο σε εμπορικούς και επιστημονικούς τομείς, αλλά και σε διάφορους εξειδικευμένους τομείς, όπως η απάντηση σε ερωτημάτων και η υποστήριξη αποφάσεων στις επιστήμες υγείας. [23]

Σύμφωνα με το Resource Description Framework (RDF), κάθε ακμή ενός Γράφου Γνώσης (KG) αναπαριστάται με μια τριάδα της μορφής *(subject, predicate, object)* (SPO), όπου *subject* και *object* είναι οντότητες και *predicate* είναι η σχέση μεταξύ τους. Ωστόσο, τέτοιου είδους τριάδες καθιστούν τους KGs δύσκολους ως προς τον χειρισμό τους. Παρόλο που οι KGs γίνονται εύκολα κατανοητοί από τους ανθρώπους και περιέχουν πλούσιες πληροφορίες για τον κόσμο, είναι δύσκολο να τους εκμεταλλευτούμε για τα εφαρμογές Μηχανικής Μάθησης (ML). Ένα παράδειγμα ενός KG φαίνεται στην Εικόνα I.1, όπου η ακμή *genre* από το *J.K. Rowling* προς το *Tragicomedy* εκφράζεται από την SPO τριάδα *(J.K. Rowling, genre, Tragicomedy)*. [2, 19, 33]

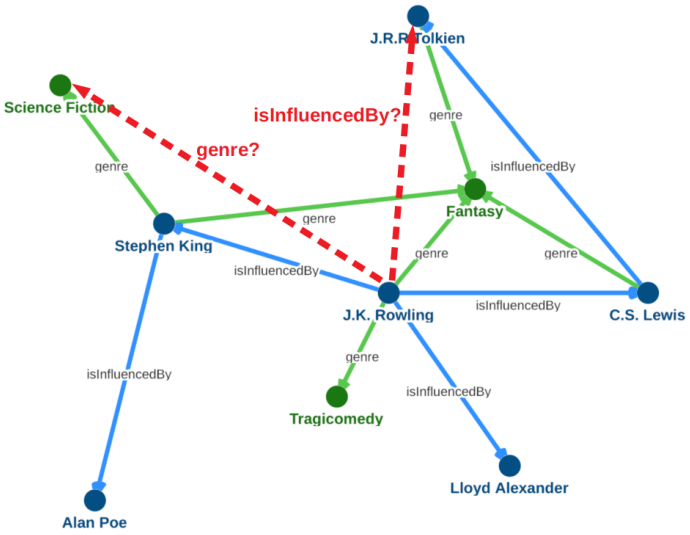


Σχήμα I.1: Παράδειγμα Γράφου Γνώσης [31]

Ένα άλλο πρόβλημα των KGs αφορά την πληρότητα τους. Παρόλο που έχουν δημιουργηθεί πολλές Βάσεις Γνώσης μεγάλης κλίμακας (KB) που περιέχουν δισεκατομμύρια γεγονότα (τριάδες), όπως η Freebase [1] και η YAGO [2], απέχουν πολύ από το πλήρες και είναι πιθανό ορισμένες από τις

[1] www.freebase.com
[2] inf.mpg.de/yago-naga/yago/

ακμές των KG να είναι λανθασμένες. Σε πολλές περιπτώσεις, τα ελλείποντα γεγονότα μπορούν να προκύψουν από τα υπάρχοντα ή πιο συχνά μπορούν να συναχθούν από ένα πιθανοτικό μοντέλο που εξαρτάται από τα υπάρχοντα γεγονότα. Επομένως, το *Link Prediction* ή το *Knowledge Graph Completion* ασχολείται με την πρόβλεψη της ύπαρξης ή της πιθανότητας ορθότητας των τριάδων σε έναν KG. Για παράδειγμα, στην Εικόνα I.2, η ακμή *genre* από το *J.K. Rowling* προς το *Science Fiction* μπορεί να προβλεφθεί, από τα γεγονότα *(J.K. Rowling, isInfluencedBy, Stephen King)* και *(Stephen King, genre, Science Fiction)*. Ομοίως, τα γεγονότα *(J.K. Rowling, genre, Fantasy)* και *(J.R.R. Tolkien, genre, Fantasy)* υποδηλώνουν την ελλείπουσα *isInfluencedBy* από το *J.K. Rowling* προς το *J.R.R. Tolkien*. [23, 28]



Σχήμα I.2: Παράδειγμα Link Prediction [31]

Στη Στατιστική Σχεσιακή Μάθηση (SRL) η είσοδος περιέχει αντικείμενα και τις σχέσεις τους, δηλαδή υα δεδομένα παίρνουν τη μορφή ενός γράφου και ο στόχος είναι η εκμάθηση από τις τριάδες που περιέχονται σε έναν KG. Η εκπαίδευση ενός μοντέλου σε ένα KG θα πρέπει να είναι σε θέση να διαχωρίζει τις σωστές τριάδες από τις λανθασμένες και να παρέχει μια κατάλληλη αναπαράσταση οντοτήτων και σχέσεων που μπορούν να χρησιμοποιηθούν σε εφαρμογές ML. Για να αντιμετωπιστούν τα ζητήματα μεγέθους και επιδόσης, οι ερευνητές αναπαριστούν τα μέρη του KG (οντότητες και τύποι σχέσεων) σε συνεχείς διανυσματικούς χώρους χαμηλών διαστάσεων, προκειμένου να απλουστευθούν οι χειρισμοί, διατηρώντας ταυτόχρονα τη δομή του KG. [23]

Οι πιο κοινοί τύποι των SRL μοντέλων είναι τα Latent Feature Models, ειδικότερα τα Translation Distance Models, τα οποία περιγράφονται λεπτομερώς στο Κεφάλαιο I.2. Το TransE, πιθανότατα το πιο αντιπροσωπευτικό μοντέλο, μετρά την αληθοφάνεια ενός γεγονότος ως την απόσταση μεταξύ δύο οντοτήτων, με απλό και αποδοτικό τρόπο. Ωστόσο, θεωρεί μόνο τις απευθείας σχέσεις μεταξύ των οντοτήτων. Αντίθετα, η επέκτασή του, PTransE, δημιουργεί ακόμη μονοπάτια σχέσεων πολλαπλών βημάτων στον KG, αλλά υστερεί στη χρονική πολυπλοκότητα. [18]

I.1.2 Συνεισφορά Διπλωματικής Διατριβής

Μία μάλλον ανεξερεύνητη προσέγγιση, με βάση τις καλύτερες γνώσεις του συγγραφέα, που υιοθετούμε σε αυτή τη διατριβή, είναι ο συνδυασμός των μοντέλων TransE και PTransE για να αντιμετωπίσουμε τα προβλήματα που αναφέρθηκαν παραπάνω. Συγκεκριμένα, επεκτείνουμε το μοντέλο TransE με την προσθήκη μονοπατιών πολλαπλών βημάτων και αντίστροφων σχέσεων στον KG, όπως και το PTransE [18], χωρίς όμως να επηρεάζεται η πολυπλοκότητα του TransE. Υπολογίζουμε την αξιοπιστία κάθε διαδρομής πριν από την εκπαίδευση και τροποποιούμε margin-based loss functions λαμβάνοντας υπόψη τις τιμές της αξιοπιστίας.

Επιπλέον, τα σύνολα δεδομένων, όπως το Freebase ή το WordNet [3], έχουν μελετηθεί σε βάθος τα τελευταία χρόνια [33]. Επομένως, εστιάζουμε στο SNOMED CT [4], το οποίο κωδικοποιεί την ιατρική ορολογία και περιγράφεται λεπτομερώς στο Κεφάλαιο I.3. Αξίζει να σημειωθεί ότι η περιγραφική λογική, στην οποία βασίζεται το περιεχόμενο του SNOMED CT, αποτελεί σημαντικό στοιχείο για την συμπλήρωση του KG.

Η λεπτομερής μεθοδολογία περιγράφεται στο Κεφάλαιο I.3, ενώ η πειραματική διαδικασία στο Κεφάλαιο I.4. Τα αποτελέσματα της μελέτης παρουσιάζονται και αναλύονται στο Κεφάλαιο I.5. Τα συμπεράσματα παρουσιάζονται στο Κεφάλαιο I.6, μαζί με μια προοπτική για περαιτέρω διερεύνηση αυτής της εργασίας.

[3] wordnet.princeton.edu/
[4] www.snomed.org/

I.2 Επιστημονικό Υπόβαθρο

I.2.1 Γράφοι Γνώσης

Όπως αναφέρθηκε στην Ενότητα I.1.2, οι KGs παίζουν έναν κεντρικό ρόλο σε πολλές εφαρμογές της Τεχνητής Νοημοσύνης (AI). Συνήθως περιέχουν τεράστια αριθμό δομημένων δεδομένων στη μορφή RDF τριάδων *(subject, predicate, object).*

Ενώ οι υπάρχοντες κωδικοποιούν αληθή γεγονότα, υπάρχουν διαφορετικές προσεγγίσεις για την ερμηνεία των μη υπαρχουσών τριάδων. Σύμφωνα με την *Closed World Assumption* (CWA), οι μη υπάρχουσες τριάδες υποδεικνύουν ψευδείς σχέσεις. Από την άλλη πλευρά, κάτω από το *Open World Assumption* (OWA), μία μη υπάρχουσα τριάδα ερμηνεύεται ως άγνωστη, δηλαδή η αντίστοιχη σχέση μπορεί να είναι είτε αληθής είτε ψευδής. Θα πρέπει να αναφερθεί ότι το RDF κάνει το OWA. [23]

Καθώς οι περισσότεροι KGs έχουν κατασκευαστεί είτε συνεργατικά, δηλαδή χειροκίνητα από μια ανοιχτή ομάδα εθελοντών ή (εν μέρει) αυτομάτως, συχνά υποφέρουν από έλλειψη πληρότητας ή ορθότητας. Επομένως, το *Link Prediction* ή το *Knowledge Graph Completion* ασχολείται με την πρόβλεψη της ύπαρξης ή της πιθανότητας ορθότητας των τριάδων σε KG.

I.2.2 Συμβολισμοί

Πριν προχωρήσουμε, παρουσιάζουμε εν συντομία τους βασικούς συμβολισμούς. Δηλώνουμε τις μεταβλητές με μικρά γράμματα - όπως a -, τα διανύσματα-στήλες (μεγέθους N) με έντονα μικρά γράμματα - όπως $\mathbf{a}$ - τους πίνακες (μεγέθους $N_1 \times N_2$) με έντονα κεφαλαία γράμματα - όπως $\mathbf{A}$ - και τους tensors (μεγέθους $N_1 \times N_2 \times N_3$) με έντονα κεφαλαία γράμματα με υπογράμμιση - όπως $\underline{\mathbf{A}}$. Δηλώνουμε (i, j, k)-ο στοιχείο με a_{ijk} (το οποίο είναι μεταβλητή). Δηλώνουμε την L_p norm ενός διανύσματος με $\|\mathbf{a}\|_p$ και τη Frobenius norm ενός πίνακα με $\|\mathbf{A}\|_F$. Το $\|\mathbf{x}\|_{1/2}$ εκφράζει είτε την L_1 norm είτε την L_2 norm. Οι μαθηματικοί ορισμοί των L_1, L_2 και Frobenius norms παρουσιάζονται στο Παράρτημα A.

Δηλώνουμε το πλήθος των οντοτήτων ενός KG με N_e, το πλήθος των τύπων σχέσεων με N_r, το πλήθος των παρατηρημένων τριάδων με $\mathcal{D}$, το πλήθος των σωστών παρατηρημένων τριάδων με $\mathcal{D}^+$ και το πλήθος των λανθασμένων παρατηρημένων τριάδων με $\mathcal{D}^-$.

Προκειμένου να οριστούν πιο τυπικά τα στατιστικά μοντέλα για KGs, εισάγουμε κάποιο μαθηματικό υπόβαθρο. Έστω $\mathcal{E} = e_1, ..., e_{N_e}$ το σύνολο όλων των οντοτήτων και $\mathcal{R} = r_1, ..., r_{N_r}$ το σύνολο όλων των τύπων σχέσεων σε έναν KG. Όλες οι δυνατές τριάδες στο $\mathcal{E} \times \mathcal{R} \times \mathcal{E}$ μπορούν να ομαδοποιηθούν σε έναν tensor τριών διαστάσεων $\underline{\mathbf{Y}} \in \{0, 1\}^{N_e \times N_e \times N_r}$, *Adjacency Tensor*, όπου το $y_{ijk} = 1$ υποδηλώνει την ύπαρξη μιας τριάδας και η ερμηνεία του $y_{ijk} = 0$ εξαρτάται από το αν γίνεται OWA ή CWA, δηλαδή:

$$y_{ijk} = \begin{cases} 1, & \text{if the triple } (e_i, r_k, e_j) \text{ exists.} \\ 0, & \text{otherwise.} \end{cases} \tag{I.1}$$

Ο Πίνακας I.1 παρουσιάζει περιληπτικά κάποιους συμβολισμούς που χρησιμοποιούνται στα επόμενα κεφάλαια.

Πίνακας I.1: Περίληψη Συμβολισμών [18, 23]

Relational Data		
Symbol	Meaning	
N_e	Number of entities	
N_r	Number of relations	
N_d	Number of training examples	
e_i	i-th entity in the dataset	
r_k	k-th relation in the dataset	
D^+	Set of observed positive triples	
D^-	Set of observed negative triples	
Latent Feature Models		
Symbol	Meaning	Size
H_e	Number of latent features for entities	
H_r	Number of latent features of relations	
e_i	Latent feature of the entity e_i	H_e
r_k	Latent feature of the relation r_k	H_r
Other Symbols		
Symbol	Meaning	Size
$\underline{\mathbf{Y}}$	Adjacency tensor	$N_e \times N_e \times N_r$
$R(p\|h,t)$	Reliability of path p given (h,t)	

I.2.3 Στατιστική Σχεσιακή Μάθηση

I.2.3.1 Latent Feature Models

Semantic Matching Models. Τα semantic matching models εκμεταλλεύονται similarity-based scoring functions. Μετρούν την αξιοπιστία των γεγονότων συνδυάζοντας την λανθάνουσα σημασιολογία των οντοτήτων και των σχέσεων που ενσωματώνονται στις αναπαραστάσεις του διανυσματικού χώρου τους. Κάποια χαρακτηριστικά μοντέλα αναλύονται στην Υποενότητα 2.3.1.

Translational Distance Models. Translational distance models εκμεταλλεύονται distance-based scoring functions. Μετρούν την αληθοφάνεια ενός γεγονότος ως την απόσταση μεταξύ των δύο οντοτήτων, συνήθως μετά από μια translation που πραγματοποιείται από τη σχέση.

TransE. Το TransE [2] είναι ένα μοντέλο βασισμένο στην ενέργεια για την εκμάθηση μικρών διαστάσεων embeddings των οντοτήτων. Οι σχέσεις αναπαριστώνται *translations in the embedding space*, έτσι ώστε με δεδομένο ένα γεγονός (h,r,t), το embedding της ουράς t να βρίσκεται κοντά στο embedding της κεφαλής h συν ένα διάνυσμα που εξαρτάται από τη σχέση r. Με άλλα λόγια, $\mathbf{h}+\mathbf{r} \approx \mathbf{t}$ όταν το (h,r,t) ισχύει, θεωρώντας ένα translation διάνυσμα $\mathbf{r}$ και τις embedded οντότητες $\mathbf{h}$ και $\mathbf{t}$. Το scoring function μιας τριάδας x_{ijk} ορίζεται ως η (αρνητική) απόσταση μεταξύ του $\mathbf{h}+\mathbf{r}$ και του $\mathbf{t}$, δηλαδή

$$f_{ijk}^{TransE} := -\|\mathbf{e_i} + \mathbf{r_k} - \mathbf{e_j}\|_{1/2} \tag{I.2}$$

Το score αναμένεται να έχει μεγάλη τιμή, αν μια τριάδα x_{ijk} ισχύει.

Η βασική ιδέα του TransE είναι ότι αν ισχύει η (h,r,t), τότε $\mathbf{h}+\mathbf{r} \approx \mathbf{t}$ και το $\mathbf{t}$ πρέπει να είναι ο κοντινότερος γείτονας του $\mathbf{h}+\mathbf{r}$, ενώ το $\mathbf{h}+\mathbf{r}$ πρέπει να βρίσκεται μακριά από το $\mathbf{t}$ σε διαφορετική περίπτωση. Για την εκμάθηση τέτοιων, ένα margin-based ranking κριτήριο ελαχιστοποιείται στο training set:

$$\mathcal{L} = \sum_{(h,r,t)\in\mathcal{D}^+} \sum_{(h',r,t')\in\mathcal{D}'_{(h,r,t)}} [\gamma + \|\mathbf{h}+\mathbf{r}-\mathbf{t}\|_{1/2} - \|\mathbf{h}'+\mathbf{r}-\mathbf{t}'\|_{1/2}]_+ \tag{I.3}$$

όπου το $[x]_+$ δηλώνει το θετικό μέρος του x, $\gamma > 0$ είναι ένα περιθώριο υπερπαραμέτρου και το ${\mathcal{D}'}_{(h,r,t)}$ είναι το σύνολο των corrupted τριάδων δεδομένου του γεγονότος (h, r, t). Το σύνολο ${\mathcal{D}'}_{(h,r,t)}$ ορίζεται ως:

$$\begin{aligned} {\mathcal{D}'}_{(h,r,t)} = \{(h', r, t) \mid h \neq h' \wedge (h, r, t) \in \mathcal{D}^+ \wedge h' \in \mathcal{E}\} \\ \cup \{(h, r, t') \mid t \neq t' \wedge (h, r, t) \in \mathcal{D}^+ \wedge t' \in \mathcal{E}\} \end{aligned} \tag{I.4}$$

Η βελτιστοποίηση πραγματοποιείται από το stochastic gradient descent. Η λεπτομερής βελτιστοποίηση από *et al.* [2] περιγράφεται στον Αλγόριθμο 1.

Αν και το μοντέλο TransE είναι απλό και αποτελεσματικό, έχει ελαττώματα στην αντιμετώπιση των 1-to-N, N-to-1 και N-to-N σχέσεων. Για παράδειγμα για μία1-to-N σχέση, δεδομένης της σχέσης r και της οντότητας h, το TransE επιβάλλει $\mathbf{h} + \mathbf{r} \approx \mathbf{t_i}$ για όλα τα $i = 1, ..., p$, τέτοια ώστε $(h, r, t_i) \in \mathcal{D}^+$, και επομένως $t_1 \approx ... \approx t_p$. Παρόμοια μειονεκτήματα υπάρχουν και για τις N-to-1 και N-to-N σχέσεις.

Η χωρική του πολυπλοκότητα είναι $\mathcal{O}(N_e H_e + N_r H_r)$ και η χρονική του πολυπλοκότητα $\mathcal{O}(H_e)$.

Για να ξεπεραστούν τα μειονεκτήματα του TransE στον χειρισμό των σχέσεων 1-to-N, N-to-1 και N-to-N, το μοντέλο TransE έχει επεκταθεί επιτρέποντας σε μια οντότητα να έχει ξεχωριστές αναπαραστάσεις όταν εμπλέκεται σε διαφορετικές σχέσεις. Μερικές από τις επεκτάσεις του παρουσιάζονται στο Κεφάλαιο 2.

PTransE. Το TransE λαμβάνει υπόψη μόνο τις άμεσες σχέσεις μεταξύ οντοτήτων. Το Path-based TransE (PTransE) [18] δημιουργεί ακόμη τριάδες χρησιμοποιώντας συνδέσεις ζευγών μέσω μονοπατιών σχέσεων. Δεδομένου ενός μονοπατιού $p = (r_1, ..., r_l)$ που ενώνει δύο οντότητες h και t, το PTransE θεωρεί τρεις τύπους λειτουργιών σύνθεσης:

- Addition (ADD): $\mathbf{p} = \mathbf{r}_1 + ... + \mathbf{r}_l$
- Multiplication (MUL): $\mathbf{p} = \mathbf{r}_1 \circ ... \circ \mathbf{r}_l$
- Recurrent Neural Network (RNN): $\mathbf{c}_i = f(\mathbf{W}[\mathbf{c}_{i-1}; \mathbf{r}_i])$, όπου το $\mathbf{c}_i$ υποδηλώνει ένα accumulated path διάνυσμα για την i-η σχέση.

Καθώς δεν είναι όλα τα μονοπάτια σχέσεων σημαντικά και αξιόπιστα για την εκμάθηση, το PTransE υπολογίζει την αξιοπιστία κάθε μονοπατιού, σύμφωνα με το path-constraint resource allocation (PCRA). Δεδομένης μια τριάδας (h, p, t), η διαδρομή ροής μπορεί να γραφτεί ως $S_0 \xrightarrow{r_1} S_1 \xrightarrow{r_2} ... \xrightarrow{r_l} S_l$, όπου $S_0 = h$ και $s_l = t$. Για κάθε μία οντότητα $m \in S_i$, το σύνολο $S_{i-1}(\cdot, m)$ περιλαμβάνει τους άμεσους "προγόνους" μέσω της σχέσης r_i στο S_{i-1}. Η αξιοπιστία του μονοπατιού p δεδομένου του (h, m) ορίζεται ως

$$R(p \mid h, m) = R_p(m) = \sum_{n \in S_{i-1}(\cdot, m)} \frac{1}{\mid S_i(n, \cdot) \mid} R_p(n) \tag{I.5}$$

όπου το σύνολο $S_i(n, \cdot)$ τους άμεσους "διαδόχους" των $n \in S_{i-1}$ ακολουθώντας τη σχέση r_i. Για κάθε ένα μονοπάτι σχέσεων p, η κεφαλή h έχει $R_p(h) = 1$.

Για κάθε ένα γεγονός $(h, r, t) \in \mathcal{D}^+$, το PTransE ορίζει ένα loss ως

$$\mathcal{L}_{path} = \frac{1}{Z} \sum_{p \in P(h,t)} R(p \mid h, t) \cdot E(h, p, t) \tag{I.6}$$

όπου το $P(h, t)$ είναι το σύνολο όλων των μονοπατιών μεταξύ των h και t, το $Z = \sum_{p \in P(h,t)} R(p \mid h, t)$ είναι ένας normalization factor και η energy function $E(h, p, t)$ ορίζεται ως

$$E(h, p, t) = \|\mathbf{h} + \mathbf{p} - \mathbf{t}\| = \|\mathbf{p} - \mathbf{r}\| \tag{I.7}$$

Επιπλέον, το PtransE προσθέτει στον KG την αντίστροφη σχέση για κάθε μία τριάδα, δηλαδή η τριάδα (t, r^{-1}, h) δημιουργείται για κάθε $(h, r, t) \in \mathcal{D}^+$.

Η χωρική του πολυπλοκότητα είναι $\mathcal{O}(N_e H_e + N_r H_r)$ και η χρονική του πολυπλοκότητα είναι $\mathcal{O}(H_e PL)$ για τους τύπους ADD και MUL και $\mathcal{O}(H_e^2 PL)$ για τον τύπο RNN, όπου P το πλήθος των μονοπατιών μεταξύ δύο οντοτήτων και L το μήκος των μονοπατιών.

I.2.3.2 Graph Feature Models

Οι προσεγγίσεις των Graph Feature Models υποθέτουν ότι η ύπαρξη μιας ακμής μπορεί να προβλεφθεί με την εξαγωγή χαρακτηριστικών από τις παρατηρημένες ακμές του γράφου. Σε αντίθεση με τα Latent Feature Models, αυτό το είδος συλλογισμών εξηγεί τις τριάδες απευθείας από τα παρατηρημένα δεδομένα στον KG. Αυτές οι μέθοδοι βασίζονται στην homophily, ενώ η ομοιότητα των οντοτήτων μπορεί να προέρχεται από τη γειτονιά των κόμβων ή από την ύπαρξη διαδρομών μεταξύ κόμβων. [23] Υποθέτουμε ότι όλα τα y_{ijk} είναι υπό όρους ανεξάρτητα, λαμβάνοντας υπόψη τα παρατηρημένα graph features και τις πρόσθετες παραμέτρους. Η ύπαρξη μιας τριάδας x_{ijk} προβλέπεται από μια score function, η οποία αντιπροσωπεύει την εμπιστοσύνη του μοντέλου για την ύπαρξη της τριάδας. Μια αναλυτικότερη περιγραφή των Graph Feature Models βρίσκεται στην Υποενότητα 2.3.2.

I.2.3.3 Markov Random Fields

Το δίκτυο Markov (γνωστό και ως Markov Random Field) είναι ένα μοντέλο για την κοινή κατανομή ενός συνόλου μεταβλητών $X = (X_1, X_2, ..., X_n) \in X$ [26]. Τα Markov Logic Networks (MLNs) είναι μια αναπαράσταση SRL ως ένα μη κατευθυνόμενο γραφικό μοντέλο. Τα γραφικά μοντέλα χρησιμοποιούν γράφους για την κωδικοποίηση εξαρτήσεων μεταξύ τυχαίων μεταβλητών. Το γεγονός αυτό είναι η κύρια διαφορά τους από τους KGs, οι οποίοι κωδικοποιούν την ύπαρξη γεγονότων. Κάθε πιθανό γεγονός y_{ijk} (ή τυχαία μεταβλητή), το οποίο μπορεί να εξαρτάται από οποιεσδήποτε $N_e \times N_e \times N_r - 1$ άλλες τυχαίες μεταβλητές, αναπαρίσταται ως κόμβος στον γράφο, ενώ κάθε εξάρτηση μεταξύ τους αναπαρίσταται ως ακμή. Τα γραφήματα αυτά ονομάζονται *dependency graphs*. [23] Τα Markov Random Fields παρουσιάζονται αναλυτικότερα στην Υποενότητα 2.3.3.

I.2.4 Αρνητική Δειγματοληψία

Ένα κοινό πρόβλημα είναι ότι οι περισσότεροι KGs περιέχουν μόνο θετικά παραδείγματα εκπαίδευσης, δηλαδή $y_{ijk} = 1$ για όλα τα $(i, j, k) \in \mathcal{D}$. Η εκπαίδευση σε μόνο θετικά δεδομένα μπορεί να είναι καταστροφική, διότι το μοντέλο μπορεί εύκολα να γενικευθεί. Ως εκ τούτου, υπάρχει η ανάγκη αρνητικής δειγματοληψίας, η δημιουργία λανθασμένων τριάδων. [8] Προτείνονται διαφορετικές προσεγγίσεις για τη δημιουργία αρνητικών γεγονότων [23], όπως:

- να θεωρήσουμε CWA και να υποθέσουμε ότι όλες οι τριάδες (με συνέπεια τύπου) που δεν βρίσκονται στο $\mathcal{D}$ είναι λανθασμένες. Για μη πλήρεις KGs αυτή η υπόθεση μπορεί να οδηγήσει σε ζητήματα μεγέθους.

- να εκμεταλλευτούμε τους περιορισμούς τύπων, λειτουργικούς ή εύρη τιμών για την εγκυρότητα ως προς την τιμή, στη δομή ενός KG. Ωστόσο, οι λειτουργικοί περιορισμοί είναι περιορισμένοι, ενώ τα αρνητικά γεγονότα με βάση τους περιορισμούς τύπων και τα έγκυρα εύρη τιμών συνήθως δεν επαρκούν για την εκπαίδευση.

- να φθείρουμε (corrupt) τις σωστές τριάδες, αντικαθιστώντας είτε τν κεφαλή είτε την ουρά με μια τυχαία οντότητα από $\mathcal{E}$. Συγκεκριμένα,

$$\begin{aligned} \mathcal{D}^- = &\{(e_l, r_k, e_j) \mid e_i \neq e_l \wedge (e_i, r_k, e_j) \in \mathcal{D}^+\} \\ &\cup \{(e_i, r_k, e_l) \mid e_j \neq e_l \wedge (e_i, r_k, e_j) \in \mathcal{D}^+\} \end{aligned} \tag{I.8}$$

- να φθείρουμε (corrupt) τις σωστές τριάδες, αντικαθιστώντας είτε τν κεφαλή είτε την ουρά με μια τυχαία οντότητα από $\mathcal{E}$ ή τη σχέση με μια τυχαία από το $\mathcal{R}$. Συγκεκριμένα,

$$\begin{aligned}\mathcal{D}^- = &\{(e_l, r_k, e_j) \mid e_i \neq e_l \wedge (e_i, r_k, e_j) \in \mathcal{D}^+\} \\ &\cup \{(e_i, r_k, e_l) \mid e_j \neq e_l \wedge (e_i, r_k, e_j) \in \mathcal{D}^+\} \\ &\cup \{(e_i, r_l, e_j) \mid r_k \neq r_l \wedge (e_i, r_k, e_j) \in \mathcal{D}^+\}\end{aligned} \tag{I.9}$$

- να θεωρήσουμε *Local-Closed World Assumption* (LCWA). Υποθέτουμε ότι ένας KG είναι μόνο *τοπικά* πλήρης. Ακριβέστερα, αν υπάρχει μια τριάδα για ένα συγκεκριμένο ζεύγος subject-predicate $(e_i, r_k$, τότε κάθε μη υπάρχουσα τριάδα $(e_i, r_k, \cdot)$ είναι πράγματι ψευδής και συμπεριλαμβάνεται στο $\mathcal{D}^-$. Η παραδοχή γίνεται μόνο για λειτουργικές σχέσεις. Αν δεν υπάρχει παρατηρημένη τριάδα για το ζεύγος (e_i, r_k), τότε όλες οι τριάδες $(e_i, r_k, \cdot)$ θεωρούνται άγνωστες και δεν περιλαμβάνονται στο $\mathcal{D}^-$.

I.2.5 Χρήσιμοι Ορισμοί

Στην ενότητα αυτή παρουσιάζονται κάποιοι ορισμοί που χρησιμοποιούνται στο κείμενο. [5, 23]

- *Knowledge Graph* παρέχουν πληροφορία με τη μορφή οντοτήτων και σχέσεων μεταξύ τους. Οι κόμβοι αντιπροσωπεύουν οντότητες, ενώ οι ακμές αντιπροσωπεύουν υπάρχουσες σχέσεις.
- *Statistical Relational Learning* ασχολείται με τη δημιουργία στατιστικών μοντέλων για σχεσιακά δεδομένα.
- *Link Prediction/Knowledge Graph Prediction* ασχολείται με την πρόβλεψη της ύπαρξης ή της πιθανότητας ορθότητας των τριάδων σε KG.
- *Embedding* είναι μια αναπαράσταση διακριτών αντικειμένων, όπως λέξεις, σε διανύσματα πραγματικών αριθμών.
- *Closed World Assumption (CWA)*, οι μη υπάρχουσες τριάδες υποδεικνύουν λανθασμένες σχέσεις.
- *Open World Assumption (OWA)*, μία μη υπάρχουσα τριάδα ερμηνεύεται ως άγνωστη, δηλαδή η αντίστοιχη σχέση μπορεί να είναι είτε αληθής είτε ψευδής.
- *Adjacency Tensor* είναι ένας tensor τριών διαστάσεων $\underline{\mathbf{Y}} \in \{0,1\}^{N_e \times N_e \times N_r}$, όπου το $y_{ijk} = 1$ ιυποδηλώνει την ύπαρξη μιας τριάδας και η ερμηνεία του $y_{ijk} = 0$ εξαρτάται από το αν γίνεται OWA ή CWA.
- *Negative Samples/Examples* είναι τα παρατηρημένα λανθασμένα γεγονότα (τριάδες) στο σύνολο δεδομένων.
- *Latent Features* είναι τα χαρακτηριστικά, τα οποία δεν παρατηρούνται άμεσα στα δεδομένα.

I.3 Μεθοδολογία

I.3.1 Προτεινόμενο Μοντέλο

Στην παρούσα εργασία εισάγουμε το μοντέλο TransEP, το οποίο είναι ένας συνδυασμός των TransE και PTransE. Στόχος μας είναι να διατηρήσουμε την αποτελεσματικότητα του TransE και να επωφεληθούμε από τις πρόσθετες γνώσεις του PTransE.

Πριν την εκπαίδευση προσθέτουμε στον KG τα μονοπάτια σχέσεων και τις αντίστροφες σχέσεις και υπολογίζουμε την αξιοπιστία κάθε διαδρομής. Επιπλέον, καθώς δεν είναι πρακτικό να απαριθμήσουμε όλα τα πιθανά μονοπάτια μεταξύ οντοτήτων χωρίς περιορισμούς, αφαιρούμε τις τριάδες με πολύ χαμηλό βαθμό αξιοπιστίας.

Επειδή διαφορετικές τριάδες έχουν διαφορετική συνεισφορά στο συνολικό πληροφοριακό περιεχόμενο του KG, το loss function θα πρέπει να λαμβάνει περισσότερο υπόψη τις τριάδες με μεγαλύτερο πληροφοριακό περιεχόμενο. Έτσι, μετά τον υπολογισμό των βαρών/αξιοπιστίας των τριάδων, τροποποιούμε το margin-based loss function του TransE και δημιουργούμε ένα weighted-margin-based loss function, όπως οι Mai *et. al* στο [20].

I.3.2 Σύνολα Δεδομένων

Σε αυτή τη διπλωματική εργασία μελετάμε δεδομένα που εξάγονται από το Freebase και το SNOMED CT. Οι στατιστικές τους παρουσιάζονται στον Πίνακα I.2.

I.3.2.1 FB15K

Το Freebase περιέχει σήμερα περίπου 1,9 δισεκατομμύρια τριάδες και εκατοντάδες ή χιλιάδες τύπους σχέσεων που σχετίζονται με την παγκόσμια γνώση και κατασκευάζεται αυτόματα ή ημιαυτόματα από διάφορους πόρους σε εκατομμύρια οντότητες. Αυτές οι σχέσεις περιλαμβάνουν τα include born-in, nationality, is-in (για τις γεωγραφικές οντότητες), part-of (για οργανισμούς, μεταξύ άλλων) και πολλά άλλα. Το σύνολο δεδομένων FB15K εισήχθη από τον Bordes *et al.* στο [2]. Πρόκειται για ένα υποσύνολο του Freebase, το οποίο περιέχει περίπου 14.951 οντότητες με 1.345 διαφορετικές σχέσεις.

I.3.2.2 SNOMED CT

Το SNOMED CT (Systematized Nomenclature of Medicine – Clinical Terms) είναι μια συστηματικά οργανωμένη ηλεκτρονική συλλογή ιατρικών όρων που παρέχουν κώδικες, όρους, συνώνυμα και ορισμούς που χρησιμοποιούνται στην κλινική τεκμηρίωση και αναφορά. Κωδικοποιεί τις έννοιες που χρησιμοποιούνται στις πληροφορίες για την υγεία και υποστηρίζει την αποτελεσματική κλινική καταγραφή δεδομένων με στόχο τη βελτίωση της περίθαλψης των ασθενών. Η ολοκληρωμένη κάλυψη του SNOMED CT περιλαμβάνει: κλινικά ευρήματα, συμπτώματα, διαγνώσεις, διαδικασίες, δομές σώματος, οργανισμούς και άλλες αιτιολογίες, ουσίες, φαρμακευτικά προϊόντα, συσκευές και δείγματα. [1]

Επιπλέον, οι έννοιες σχετίζονται μεταξύ τους με βάση τις σχέσεις και τα Web Ontology Language (OWL) [21] αξιώματα που παρέχουν έναν επίσημο λογικό ορισμό των εννοιών. Με άλλα λόγια, το SNOMED CT περιέχει ένα $\mathcal{TBOX}$ από ένα σύνολο αξιωμάτων. Η μεταβατικότητα και οι αλυσίδες ιδιοτήτων, οι οποίες κατά κάποιο τρόπο είναι παρόμοιες με την μεταβατικότητα αλλά περιλαμβάνουν περισσότερα από ένα χαρακτηριστικά, είναι κάποια κοινά αξιώματα και μπορούν να βελτιώσουν την ταξινόμηση.

Μια αναλυτικότερη περιγραφή του SNOMED CT και της διαδικασίας κατασκευής του KG του βρίσκεται στην Υποενότητα 4.1.2.

Πίνακας I.2: Στατιστικά των Σύνολων Δεδομένων

	FB15K	SNOMED
Number of Entities	$14,951$	$466,612$
Number of Relationships	$1,345$	113
Number of Train Data	$483,142$	$1,430,419$
Number of Valid Data	$50,000$	$357,605$
Number of Test Data	$59,071$	$447,007$

I.3.3 Κατασκευή Γράφων Γνώσης με αντίστροφες σχέσεις και μονοπάτια πολλαπλών βημάτων

I.3.3.1 Αντίστροφες Σχέσεις

Προσθέτουμε στους KGs και των δύο σύνολων δεδομένων την αντίστροφη σχέση για κάθε ένα παρατηρημένο γεγονός, δηλαδή για κάθε τριάδα (h, r, t) κατασκευάζουμε μία νέα (t, r^{-1}, h).

I.3.3.2 Μονοπάτια Πολλαπλών Βημάτων

FB15K Δεδομένου ότι το FB15K περιέχει χιλιάδες τύπους σχέσεων που σχετίζονται με την παγκόσμια γνώση χωρίς κάποια συγκεκριμένη θεματική, θεωρούμε μονοπάτια σχέσεων πολλαπλών βημάτων μεταξύ δύο οντοτήτων e_1 και e_2 μόνο αν υπάρχει τουλάχιστον ένα παρατηρημένο γεγονός (e_1, r, e_2), όπου $r \in \mathcal{R}$. Κατασκευάζουμε τις διαδρομές πολλαπλών βημάτων σύμφωνα με τον Αλγόριθμο 2.

SNOMED CT Το SNOMED CT περιέχει μόνο λίγους τύπους σχέσεων, αλλά πολλά αξιώματα OWL. Αυτό υποδηλώνει την ύπαρξη σχέσεων μεταξύ δύο οντοτήτων e_1 και e_2 χωρίς κανένα παρατηρημένο γεγονός (e_1, r, e_2). Έτσι, θεωρούμε μονοπάτια σχέσεων πολλαπλών βημάτων μεταξύ όλων των ζευγών οντοτήτων. Επιπλέον, λαμβάνουμε υπόψη την μεταβατική ιδιότητα των τύπων σχέσεων ISA και $PART\_OF$. Η κατασκευή μονοπατιών σχέσεων πολλαπλών βημάτων περιγράφεται στον Αλγόριθμο 3.

I.3.4 Αξιοπιστία Μονοπατιών

Όπως αναφέρθηκε και στην περιγραφή του PTransE, δεν είναι όλα τα μονοπάτια σχέσεων σημαντικά και αξιόπιστα για εκμάθηση. Ως εκ τούτου, υπολογίζουμε τον βαθμό αξιοπιστίας κάθε μονοπατιού σύμφωνα με την ακόλουθη εξίσωση:

$$R(p \mid h, m) = R_p(m) = \sum_{n \in S_{i-1}(\cdot, m)} \frac{1}{\mid S_i(n, \cdot) \mid} R_p(n), \tag{I.10}$$

η οποία χρησιμοποιείται επίσης από το PTransE και περιγράφεται αναλυτικά στην Υποενότητα 2.3.1. Αφού υπολογίσουμε όλα τα scores, τα διαιρούμε με το $Z = \sum_{p \in P(h,t)} R(p \mid h, t)$, τον normalization factor του PTransE. Συνεπώς, το πεδίο τιμών του reliability score είναι $(0, 1]$.

Υπολογίζουμε την αξιοπιστία κάθε διαδρομής πριν από την εκπαίδευση, ώστε να μην αυξηθεί η χρονική πολυπλοκότητα του TransE.

Σε KGs που περιέχουν μόνο άμεσες σχέσεις, υπολογίζουμε τον βαθμό αξιοπιστίας σύμφωνα με την εξίσωση I.10. Από την άλλη πλευρά, για την εκπαίδευση σε KGs που περιέχουν και αντίστροφες σχέσεις και μονοπάτια πολλαπλών βημάτων, θεωρούμε την βαθμολογία αξιοπιστίας των άμεσων μονοπατιών ως 1.0, ενώ την αξιοπιστία όλων των άλλων μονοπατιών σύμφωνα με την Εξίσωση I.10, επειδή τα παρατηρημένα γεγονότα είναι πιθανότερο να έχουν περισσότερη σημασία για την εκμάθηση.

Συνήθως υπάρχουν μεγάλες ποσότητες σχέσεων και γεγονότων για κάθε ζεύγος οντοτήτων και δεν θα ήταν πρακτικό να απαριθμήσουμε όλα τα πιθανά μονοπάτια σχέσεων μεταξύ αυτών. Για λόγους υπολογιστικής απόδοσης, σε ορισμένες περιπτώσεις θεωρούμε μόνο αυτές τις διαδρομές σχέσεων με το βαθμό αξιοπιστίας μεγαλύτερο από 0.01.

I.3.5 Loss Function

Ορίζουμε τον βαθμό αξιοπιστίας ενός μονοπατιού $h \xrightarrow{r} t$ by $w_{(h,r,t)}$. Για την εκμάθηση των KG embeddings, χρησιμοποιούμε ένα margin-based loss function, όπως και το TransE. Ωστόσο, σύμφωνα με την ιδέα των Mai *et. al* στο [20], στο loss function πολλαπλασιάζουμε με το $w_{(h,r,t)}$ την αφαίρεση τη διαφορά ανάμεσα στο scoring function, βλ. I.2, της τριάδα; (h, r, t) και του scoring function της corrupted τριάδας (h', r, t'):

$$\mathcal{L} = \sum_{(h,r,t)\in\mathcal{D}^+} \sum_{(h',r,t')\in\mathcal{D'}_{(h,r,t)}} [\gamma + w_{(h,r,t)} \cdot (\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_{1/2} - \|\mathbf{h'} + \mathbf{r} - \mathbf{t'}\|_{1/2})]_+ \tag{I.11}$$

Όπως αναφέρθηκε στην Υποενότητα I.3.1, το loss function λαμβάνει περισσότερο υπόψη του τις τριάδες που περιέχουν σημαντικότερο πληροφοριακό περιεχόμενο.

I.3.6 Αρνητική Δειγματοληψία

Για αρνητική δειγματοληψία, ακολουθούμε την ιδέα που περιγράφεται στην εξίσωση I.8, όπου για την κατασκευή αρνητικών τριάδων αντικαθιστούμε τυχαία είτε την κεφαλή είτε την ουρά με μια τυχαία οντότητα από το σύνολο $\mathcal{E}$.

I.4 Υλοποίηση

Η προσέγγισή μας, το TrasnEP, αξιολογείται σε δεδομένα που εξάγονται από το Freebase και το SNOMED CT που περιγράφονται λεπτομερώς στις Υποενότητες 4.1.1 και 4.1.2 αντίστοιχα.

Για τα πειράματά μας, χρησιμοποιούμε το GitHub repository [11] των Han *et. al* [10] υλοποιημένο με PyTorch. Χρησιμοποιήσαμε την υλοποίησή τους για το TransE και την τροποποιήσαμε για να υλοποιήσουμε το TransEP.

I.4.1 Πρωτόκολλο Αξιολόγησης

Για αξιολόγηση, χρησιμοποιείται η ίδια διαδικασία κατάταξης όπως στο [3]. Για κάθε τριάδα του test set, η κεφαλή αφαιρείται και αντικαθίσταται από κάθε μία από τις οντότητες στο $\mathcal{E}$. Τα scoring functions αυτών των αλλοιωμένων τριάδων υπολογίζονται πρώτα από τα μοντέλα και στη συνέχεια ταξινομούνται κατά αύξουσα σειρά - τελικά αποθηκεύεται η θέση της σωστής οντότητας. Όλη αυτή η διαδικασία επαναλαμβάνεται με την αφαίρεση της ουράς αντί της κεφαλής.

I.4.1.1 Μετρικές

Για όλες τις N_{test} τριάδες του test set, χρησιμοποιούμε τέσσερις μετρικές για την αξιολόγηση:

Mean Rank (MR) Η τιμή της μέσης κατάταξης. Όσο μικρότερη, τόσο το καλύτερο. Το MR υπολογίζεται από:

$$MR = \frac{1}{2N_{test}} \sum_{i=1}^{N_{test}} (rank_{ih} + rank_{it}) \tag{I.12}$$

όπου το $rank_{ih}$ και το $rank_{ih}$ αναφέρονται στη θέση της πρόβλεψης της $i_{\eta\varsigma}$ τριάδας στην κατάταξη, αντικαθιστώντας την κεφαλή ή την ουρά αντίστοιχα.

Mean Reciprocal Rank (MRR) Ο μέσος όρος όλων των αμοιβαίων θέσεων. Όσο μεγαλύτερος, τόσο το καλύτερο. Το MRR υπολογίζεται από:

$$MRR = \frac{1}{2N_{test}} \sum_{i=1}^{N_{test}} (\frac{1}{rank_{ih}} + \frac{1}{rank_{it}}) \tag{I.13}$$

όπου το $rank_{ih}$ και το $rank_{ih}$ έχουν την ίδια ερμηνεία όπως στο MR. [32]

Hits at k (Hits@k) Ο ρυθμός των σωστών οντοτήτων που εμφανίζονται στις k πρώτες θέσεις της κατάταξης. Όσο υψηλότερος, τόσο το καλύτερο. Όπως περιγράφεται από τον Bordes *et. al* [2], το Hits@k υπολογίζεται από:

$$Hits@k = \frac{1}{2N_{test}} \sum_{i=1}^{N_{test}} (I_k(rank_{ih}) + I_k(rank_{it})) \tag{I.14}$$

$$I_k(rank_i) = \begin{cases} 1, & \text{if } rank_{\leq} k, \\ 0, & \text{otherwise.} \end{cases} \tag{I.15}$$

Triple Classification Accuracy Ο ρυθμός των σωστών προβλέψεων που έγιναν στο test set. Η ακρίβεια υπολογίζεται από:

$$Accuracy = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \tag{I.16}$$

Με βάση το valid set υπολογίζουμε ένα όριο για τα scores των τριάδων. Τα σωστά δείγματα έχουν score μικρότερο από το όριο, ενώ τα αρνητικά έχουν score μεγαλύτερο από το όριο.

I.4.1.2 Φιλτράρισμα

Οι μετρικές που περιγράφηκαν παραπάνω είναι ενδεικτικές, αλλά ενδέχεται να έχουν ελαττώματα όταν κάποιες corrupted τριάδες καταλήγουν να είναι σωστές, από το training set για παράδειγμα. Σε αυτή την περίπτωση, αυτές μπορεί να κατατάσσονται πάνω από την test τριάδα, αλλά αυτό δεν θα πρέπει να θεωρείται ως σφάλμα, καθώς και οι δύο τριάδες είναι ορθές. Για να αποφευχθεί μια τέτοια παραπλανητική συμπεριφορά, ο Bordes *et al.* στο [2] πρότεινε να αφαιρέσει από τη λίστα των corrupted τριάδων όσες εμφανίζονται είτε στο training, validation ή test set (εκτός από την τριάδα που μας ενδιαφέρει). Αυτό εξασφαλίζει ότι όλες οι corrupted τριάδες δεν ανήκουν στο σύνολο δεδομένων. Στα επόμενα, τα mean ranks και hits@k υπολογίζονται σύμφωνα με δύο ρυθμίσεις: η αρχική ρύθμιση *raw* και η φιλτραρισμένη *filter*.

I.4.1.3 Περιορισμοί Τύπων

Οι Han *et. al* στο [10] δημιουργούν ένα αρχείο με τους περιορισμούς τύπων, το οποίο περιέχει περιορισμούς τύπων για κάθε σχέση, δηλαδή ποιες οντότητες κάθε σχέση έχει ως οντότητες κεφαλής και ποιες ως ουρά. Σύμφωνα με αυτό το αρχείο, για κάθε test τριάδα, η κεφαλή ή η ουρά αφαιρείται και αντικαθίσταται μόνο από οντότητες που περιέχονται στους περιορισμούς τύπων κάθε σχέσης. Στα επόμενα, τα mean ranks και hits@k υπολογίζονται επίσης σύμφωνα με τις ρυθμίσεις περιορισμών τύπων.

I.4.2 Ρύθμιση Πειραμάτων

Για τα πειράματα με το TransE και το TranEP, επιλέξαμε τον ρυθμό μάθηση λ για τον stochastic gradient descent μεταξύ των $\{0.001, 0.01, 0.1\}$, το περιθώριο γ μεταξύ των $\{1, 2\}$ και τη διάσταση των embeddings k μεταξύ των $\{50, 100\}$ σύμφωνα με το validation set κάθε συνόλου δεδομένων. Το μέτρο ανομοιότητας d ορίστηκε στην L_1 norm. Οι βέλτιστες ρυθμίσεις για τα δύο σύνολα δεδομένων, FB15K και SNOMED CT, ήταν: $k = 100$, $\lambda = 0.001$, $\gamma = 1$, και $d = L_1$. Και για τα δύο σύνολα δεδομένων, ο χρόνος εκπαίδευσης περιορίστηκε στις 1.000 εποχές. Για το FB15K, το validation εφαρμόζεται σε κάθε 10_{η} εποχή, ενώ για το SNOMED CT κάθε εποχή 100_{η}, λόγω του μεγάλου αριθμού οντοτήτων. Τα βέλτιστα μοντέλα επιλέχθηκαν με early stopping χρησιμοποιώντας τις μέσες προβλεπόμενες κατατάξεις στα validation sets (*raw* setting).

I.4.3 Πειράματα

Πραγματοποιήσαμε πειράματα με τα μοντέλα TransE και TransEP στα σύνολα δεδομένων FB15K και SNOMED CT. Για πειράματα και στα δύο σύνολα, εκπαιδεύσαμε και τα δύο μοντέλα σε τρεις KGs: (1) που περιέχουν μόνο άμεσες σχέσεις, (2) που περιέχουν άμεσες και αντίστροφες σχέσεις και μονοπάτια πολλαπλών βημάτων μήκους 2 και (3) μονοπάτια πολλαπλών βημάτων μήκους 2 και 3 και αφαίρεση τριάδων με βαθμό αξιοπιστίας μικρότερο από το κατώτατο όριο. Καθώς στο FB15K υπήρχαν πολλά γεγονότα με πολύ χαμηλό βαθμό αξιοπιστίας, εκπαιδεύσαμε τα μοντέλα και σε έναν τέταρτο KG (4) που είναι σαν τον δεύτερο, αλλά χωρίς τριάδες με βαθμό αξιοπιστίας χαμηλότερο από το κατώτατο όριο. Περιορίζουμε το μήκος των μονοπατιών σε 3 βήματα, όχι μόνο για λόγους υπολογιστικής αποδοτικότητας, αλλά και επειδή δεν υπήρξε βελτίωση στα αποτελέσματα.

I.5 Αξιολόγηση

Σε αυτό το κεφάλαιο θα χρησιμοποιήσουμε τις παραμέτρους που επιλέξαμε στην Υποενότητα I.4.2 για να πραγματοποιήσουμε τα πειράματα που περιγράφονται στην Υποενότητα I.4.3. Στους παρακάτω πίνακες παρουσιάζουμε τις επιδόσεις των TransE και TransEP (1) μόνο με άμεσες σχέσεις, (2) με αντίστροφες σχέσεις και μονοπάτια πολλαπλών βημάτων (step-k) και (3) χωρίς τριάδες με βαθμό αξιοπιστίας χαμηλότερη από το κατώτατο όριο (step-k-thres.).

I.5.1 Αποτελέσματα στο FB15K

Οι πίνακες I.3, I.4 και I.5 παρουσιάζουν τα αποτελέσματα στο FB15K για όλες τις μεθόδους που αξιολογήθηκαν, βλ. I.4.1.

Πίνακας I.3: FB15K αποτελέσματα χωρίς περιορισμούς τύπων

Method	MRR(%)		MR		Hits@10(%)		Hits@3(%)		Hits@1(%)	
	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter
TransE	**24.79**	**25.8**	**219**	**203**	**50.33**	**51.97**	**28.63**	**30.14**	**12.9**	**13.54**
TransE (step-2)	18.26	19.0	234	217	48.8	50.37	23.77	25.16	3.86	4.1
TransE (step-2-thres.)	18.9	19.7	228	211	48.64	50.23	24.24	25.63	4.78	5.1
TransE (step-3-thres.)	14.4	14.95	242	226	42.83	44.36	17.87	18.91	1.45	1.51
TransEP	23.61	24.5	244	228	48.43	49.95	26.95	28.4	12.09	12.63
TransEP (step-2)	18.37	19.07	254	238	46.28	47.78	22.63	23.84	5.31	5.58
TransEP (step-2-thres.)	19.79	20.6	245	229	47.56	49.07	24.17	25.48	6.79	7.19
TransEP (step-3-thres.)	15.41	15.99	254	237	41.98	43.39	18.47	19.49	3.29	3.47

Πίνακας I.4: FB15K αποτελέσματα με περιορισμούς τύπων

Method	MRR(%)		MR		Hits@10(%)		Hits@3(%)		Hits@1(%)	
	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter
TransE	26.61	27.68	**180**	**163**	51.8	53.5	30.21	31.77	14.97	15.68
TransE (step-2)	**27.53**	**28.57**	185	169	**52.52**	**54.12**	**31.1**	**32.65**	**16.05**	**16.75**
TransE (step-2-thres.)	26.64	27.71	184	168	51.79	53.44	30.18	31.74	15.02	15.77
TransE (step-3-thres.)	25.31	26.29	187	171	49.44	51.02	28.49	29.94	14.12	14.78
TransEP	25.73	26.67	189	173	50.34	51.92	28.94	30.44	14.4	15.02
TransEP (step-2)	26.56	27.51	193	177	50.84	52.37	29.59	30.98	15.45	16.09
TransEP (step-2-thres.)	25.64	26.63	192	176	50.56	52.15	28.77	30.21	14.87	14.94
TransEP (step-3-thres.)	24.46	25.36	194	176	47.65	49.45	27.4	28.69	13.56	14.16

Δυστυχώς, παρατηρούμε ότι το μοντέλο μας TransEP δε βελτιώνει τα αποτελέσματα στο FB15K. Το μοντέλο TransE σε KG με μόνο άμεσες σχέσεις παρουσιάζει την καλύτερη απόδοση, αν δε λάβουμε υπόψη τους περιορισμούς τύπων, σε όλες τις μετρικές. Ωστόσο, τα αποτελέσματα με τους περιορισμούς τύπων του TransE σε KG με αντίστροφες σχέσεις και μονοπάτια σχέσεων μέχρι μήκους 2 βελτιώνουν τις τιμές του *hits@k* και του Mean Reciprocal Rank, αλλά το TransE στον αρχικό KG έχει χαμηλότερο Mean Rank σε όλες τις περιπτώσεις. Επιπλέον, το TransE step-2 έχει καλύτερη απόδοση στο triple classification accuracy.

Το TransE και το TransEP στην εξέταση των διαδρομών σχέσεων με το πολύ 3 βήματα επιτυγχάνουν χειρότερα αποτελέσματα. Αυτό υποδηλώνει ότι μάλλον δεν είναι απαραίτητο να επεκτείνουμε παραπάνω το μήκος των μονοπατιών.

Πίνακας I.5: FB15K Triple Classification Accuracy

Method	Triple Classification Accuracy
TransE	84.01
TransE (step-2)	**85.27**
TransE (step-2-thres.)	84.96
TransE (step-3-thres.)	83.66
TransEP	83.17
TransEP (step-2)	83.82
TransEP (step-2-thres.)	84.26
TransEP (step-3-thres.)	82.74

I.5.2 Αποτελέσματα στο SNOMED CT

Οι ακόλουθοι πίνακες I.6, I.7 και I.8 παρουσιάζουν τα αποτελέσματα στο SNOMED CT για όλες τις μετρικές. Παρατηρούμε ότι: (1) η εκπαίδευση σε KG, ο οποίος περιλαμβάνει αντίστροφες σχέσεις και μονοπάτια σχέσεων με μέγιστο μήκος 2, είναι πιο αποδοτική σε σχέση με αυτή σε KG με μόνο άμεσες σχέσεις. (2) Η *filtered* ρύθμιση στα αποτελέσματα του TransEP δεν παρέχει σημαντική βελτίωση, σε αντίθεση με το μοντέλο TransE.

Από τον πίνακα I.6 παρατηρούμε ότι: (1) το TransE step-2 έχει καλύτερη επίδοση στο *hits@k* και Mean Reciprocal Rank. (2) Το TransEP step-2 έχει καλύτερα αποτελέσματα στο Mean Rank. (3) Τα TransEP και TransE step-2 επιτυγχάνουν συγκρίσιμα αποτελέσματα στην *raw* ρύθμιση.

Από τους πίνακες I.7 και I.8 παρατηρούμε ότι: (1) η *raw* ρύθμιση στο TransEP step-2 παρέχει σημαντική βελτίωση σε όλες τις μετρικές. (2) Η *filtered* ρύθμιση στο TransE βελτιώνει σημαντικά τα αποτελέσματα σε σχέση με το TransEP. (3) Το TransE step-2 έχει καλύτερη απόδοση για όλες τις *filtered* μετρικές. (4) Το TransEP step-2 επιτυγχάνει το υψηλότερο triple classification accuracy, αλλά συγκρίσιμο με αυτό τωνTransE step-2, step-3 και TransEP step-3.

Όπως και στο FB15K, τα TransE και TransEP σε KGs με το πολύ 3-step μονοπάτια δε βελτιώνει τα αποτελέσματα και δε επεκτείνουμε περαιτέρω το μήκος των μονοπατιών.

Πίνακας I.6: SNOMED CT αποτελέσματα χωρίς περιορισμούς τύπων

Method	MRR(%)		MR		Hits@10(%)		Hits@3(%)		Hits@1(%)	
	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter
TransE	13.37	17.71	13615	13145	25.21	27.96	15.75	19.56	7.37	12.36
TransE (step-2)	**13.93**	**22.63**	11721	11118	**27.46**	**35.84**	**15.87**	**25.14**	**7.41**	**15.71**
TransE (step-3-thres.)	10.74	21.02	12986	12283	23.05	34.1	11.8	23.17	4.92	14.35
TransEP	10.21	10.5	14007	13901	20.83	21.11	11.76	12.12	4.84	5.09
TransEP (step-2)	13.67	14.21	**10902**	**10796**	26.9	27.33	15.47	16.08	7.23	7.77
TransEP (step-3-thres.)	10.72	11.06	12374	12267	22.62	23.0	11.6	12.0	5.13	5.42

Πίνακας I.7: SNOMED CT αποτελέσματα με περιορισμούς τύπων

Method	MRR(%)		MR		Hits@10(%)		Hits@3(%)		Hits@1(%)	
	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter
TransE	13.68	18.12	6587	6117	25.54	28.42	16.07	19.99	7.66	12.75
TransE (step-2)	14.05	**22.96**	5298	**4695**	27.62	**36.24**	15.99	**25.47**	7.49	**16.01**
TransE (step-3-thres.)	10.84	21.26	5682	4978	23.23	34.51	11.93	23.51	4.98	14.53
TransEP	11.3	11.62	6131	6024	22.14	22.44	13.01	13.4	5.76	6.05
TransEP (step-2)	**15.82**	16.35	**4928**	4823	**28.78**	29.19	**17.65**	18.24	**9.5**	10.02
TransEP (step-3-thres.)	10.81	11.15	5300	5194	22.79	23.17	11.68	12.09	5.17	5.46

Πίνακας I.8: SNOMED CT Triple Classification Accuracy

Method	**Triple Classification Accuracy**
TransE	86.51
TransE (step-2)	89.67
TransE (step-3-thres.)	89.04
TransEP	84.9
TransEP (step-2)	**89.99**
TransEP (step-3-thres.)	89.17

I.6 Συμπέρασμα και Μελλοντικά Έργα

I.6.1 Περίληψη

Σε αυτή τη διπλωματική εργασία, εισαγάγαμε ένα νέο Translational Distance Model, το TransEP. Το κίνητρό μας ήταν η πρόβλεψη της ύπαρξης ή της πιθανότητας ορθότητας των τριάδων σε έναν KG, δηλαδή το Link Prediction. Συνδυάσαμε τα μοντέλα TransE και PTransE με την προσθήκη διαδρομών πολλαπλών βημάτων και αντίστροφων σχέσεων στον KG, χωρίς όμως να επηρεάζεται η πολυπλοκότητα του TransE. Επιπλέον, τροποποιήσαμε μια a margin-based loss function λαμβάνοντας υπόψη το βαθμό αξιοπιστίας κάθε σχέσης ή διαδρομής προκειμένου να λάβουμε περισσότερο υπόψη τις τριάδες με περισσότερη και σημαντική πληροφορία.

Αξιολογήσαμε τα μοντέλα TransE και TransEP σε δύο σύνολα δεδομένων, FB15K και SNOMED CT. Πραγματοποιήσαμε πειράματα σε KG όχι μόνο με άμεσες σχέσεις, αλλά και με αντίστροφες σχέσεις και μονοπάτια πολλαπλών βημάτων.

Όσον αφορά τα αποτελέσματα στο FB15K, οι διαδρομές πολλαπλών βημάτων δεν βελτιώνουν σημαντικά την απόδοση, ειδικά στο TransEP, σε σύγκριση με το αρχικό σύνολο δεδομένων. Πιθανώς, αυτό οφείλεται στο γεγονός ότι το FB15K περιέχει χιλιάδες τύπους σχέσεων που σχετίζονται με την παγκόσμια γνώση χωρίς κάποια συγκεκριμένη θεματική. Επομένως, τα μονοπάτια σχέσεων ενδέχεται να μην παρέχουν χρήσιμες πληροφορίες.

Από την άλλη πλευρά, σχετικά με το SNOMED CT, η εκπαίδευση σε KG με μονοπάτια πολλαπλών βημάτων έχει καλύτερη απόδοση από εκείνη σε KG με μόνο άμεσες σχέσεις. Τόσο το TransE όσο και το μοντέλο μας, το TransEP, επιτυγχάνουν υψηλότερες επιδόσεις σε KG με μονοπάτια μήκους δύο και αντίστροφες σχέσεις, ενώ το TransEP παρέχει καλύτερα αποτελέσματα σε κάποιες μετρικές. Αυτή η σημαντική βελτίωση πιθανόν οφείλεται στα αξιώματα OWL, στα οποία βασίζεται το SNOMED CT, όπως οι ιδιότητες των σχέσεων, ιδιαίτερα η μεταβατική ιδιότητα, που υποδηλώνουν νέα σημαντικά γεγονότα.

Και στα δύο σύνολα δεδομένων που εξετάζουν τις διαδρομές σχέσεων με το πολύ 3 βήματα τα αποτελέσματα είναι χειρότερα και, επομένως, δεν θεωρούμε μεγαλύτερες διαδρομές σχέσεων.

I.6.2 Μελλοντική Δουλειά

Υπάρχουν πολλοί τρόποι για να διερευνήσουμε τη μελλοντική δουλειά, αλλά θα μπορούσε κυρίως να πραγματοποιηθεί σε δύο κύριες κατευθύνσεις:

- Θα μπορούσαμε να χρησιμοποιήσουμε μια άλλη μέθοδο για να αξιολογήσουμε πόσο σημαντική είναι μια διαδρομή αντί αυτής του PTransE. Για παράδειγμα, μια πιθανή μέθοδος θα μπορούσε να είναι η χρήση του αλγόριθμου PageRank [4], η οποία δίνει κάποια προσέγγιση της σημασίας ή της ποιότητας μιας διαδρομής και μετρά την μεταβατική επιρροή ή τη συνδεσιμότητα των κόμβων σε έναν γράφο.

- Καθώς το TransE και, κατά συνέπεια, το TransEP έχουν ελαττώματα στον χειρισμό των σχέσεων 1-to-N, N-to-1 και N-to-N, θα μπορούσαμε να βασιστούμε σε ένα άλλο transnational distance model. Για παράδειγμα, το TransH ξεπερνά αυτά τα μειονεκτήματα και έχει την ίδια χρονική πολυπλοκότητα με το TransE. Θα μπορούσαμε να προσθέσουμε στον KG τα μονοπάτια πολλαπλών ημάτων και τις αντίστροφες σχέσεις, να υπολογίσουμε την αξιοπιστία κάθε διαδρομής και να τροποποιήσουμε το loss function του TransH σύμφωνα με τα αποτελέσματα αξιοπιστίας.

Chapter 1

Introduction

1.1 Motivation

Knowledge Graphs (KGs) provide structured information in the form of entities and relationships between them. Nodes represent entities, while edges represent existing relationships. KGs are a useful tool for knowledge representation and are used not only in commercial and scientific fields, but also in a variety of specialized domains, such as question answering and decision support in life sciences. [23]

According to the Resource Description Framework (RDF) each edge of a Knowledge Graph (KG) is represented as a triple of the form *(subject, predicate, object)* (SPO), where *subject* and *object* are entities and *predicate* is the relation between them. However, such triples make KGs hard to manipulate. Although KGs are easily comprehensible by humans and contain rich information about the world, it is difficult to take advantage of them for Machine Learning (ML) tasks. An example of a KG is shown in Figure 1.1, where the edge *genre* from *J.K. Rowling* to *Tragicomedy* is expressed by the SPO triple *(J.K. Rowling, genre, Tragicomedy)*. [2, 19, 33]

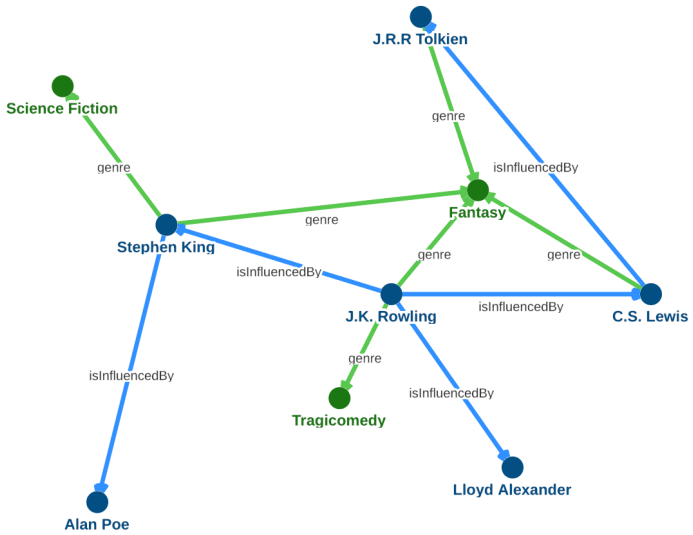


Figure 1.1: Knowledge Graph Example [31]

Another problem of KGs concerns their completeness. Even though many large-scale Knowledge Bases (KBs) containing billions of facts (triples) have been built, such as Freebase[1] and YAGO[2], they

[1] www.freebase.com
[2] www.mpi-inf.mpg.de/yago-naga/yago/

are far from complete and it is possible that some of the contained edges of the KGs are incorrect. In many cases, the missing facts can be entailed by the existing ones or more often they can be inferred by a probabilistic model conditioned on the existing facts. Therefore, *Link Prediction* or *Knowledge Graph Completion* is concerned with predicting the existence or probability of correctness of triples in KG. For instance, in Figure 1.2, the edge *genre* from *J.K. Rowling* to *Science Fiction* could be predicted, because of the facts *(J.K. Rowling, isInfluencedBy, Stephen King)* and *(Stephen King, genre, Science Fiction)*. Similarly, the facts *(J.K. Rowling, genre, Fantasy)* and *(J.R.R. Tolkien, genre, Fantasy)* could predict the missing edge *isInfluencedBy* from *J.K. Rowling* to *J.R.R. Tolkien*. [23, 28]

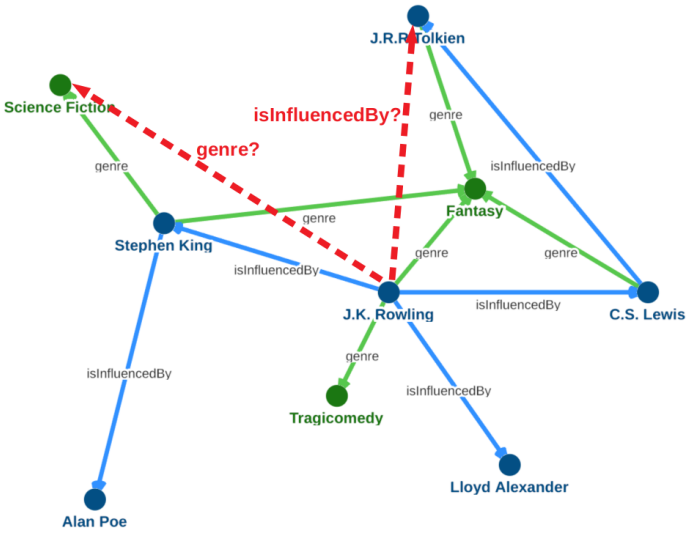


Figure 1.2: Link Prediction Example [31]

In Statistical Relational Learning (SRL) the input contains objects and their relations, i.e. data take the form of a graph, and the aim is to learn from the triples contained in a KG. A training of a model on a KG should be able to separate correct triples from false ones and provide a suitable representation of entities and relations that can be used in ML tasks. To address scalability and performance issues, researchers embed KG components (entities and relation types) into low dimensional continuous vector spaces in order to simplify the manipulation, while preserving the inherent structure of the KG. [23]

The most common types of SRL models are the Latent Feature Models, especially the Translation Distance Models, which are described in detail in Chapter 2. TransE, probably the most representative model, measures the plausibility of a fact as the distance between two entities in a simple and efficient way. Nevertheless, it considers only direct relations between entities. On the other hand, its extension, PTransE, includes also multiple-step relation paths, but it has flaws in time complexity. [18]

1.2 Scope

A rather unexplored approach, to the best of the author's knowledge, which we adopt in this thesis, is to combine TransE and PTransE in order to tackle the problems mentioned above. In particular, we extend the TransE model by adding multiple-steps paths and reverse relations to the KG,

like PTransE [18], but without impairing the time complexity of TransE. We calculate the reliability of each path before training and modify margin-based loss functions by taking under consideration the reliability scores.

Furthermore, datasets, such as Freebase or WordNet[3], have been studied intensively in recent years [33]. Therefore, we focus on SNOMED CT[4], which encodes the healthcare terminology and is described thoroughly in Chapter 4. It is worth noting that the description logic, on which SNOMED CT content is based, is an important asset for the Knowledge Graph Completion.

The detailed methodology is described in Chapter 4, while the experimental setup in Chapter 5. The results of our study are presented and discussed in Chapter 6. The conclusions are presented in Chapter 7, along with an outlook on further investigation and perspectives of this work.

[3] wordnet.princeton.edu/
[4] www.snomed.org/

Chapter 2

Scientific Background

2.1 Knowledge Graphs

As mentioned in Section 1.1, KGs play a pivotal role in many Artificial Intelligence (AI) applications. They usually contain huge amounts of structured data as the form of RDF triples *(subject, predicate, object)*.

While existing triples encode true facts, there are different paradigms for the interpretation of non-existing triples. Under the *Closed World Assumption* (CWA), non-existing triples indicate false relationships. On the other hand, under *Open World Assumption* (OWA), a non-existing triple is interpreted as unknown, i.e. the corresponding relationship can be either true or false. It should be mentioned that RDF makes the OWA.[23]

As most KGs have been built either collaboratively, i.e. manually by an open group of volunteers, or (partly) automatically, they often suffer from incompleteness or incorrectness. Therefore, *Link Prediction* or *Knowledge Graph Completion* is concerned with predicting the existence or probability of correctness of triples in KG.

2.2 Notations

Before proceeding, we briefly introduce the basic notations. We denote scalars by lower case letters, such as a; column vectors (of size N) by bold lower case letters, such as $\mathbf{a}$; matrices (of size $N_1 \times N_2$) by bold upper case letters, such as $\mathbf{A}$; and tensors (of size $N_1 \times N_2 \times N_3$) by bold upper case letters with an underscore, such as $\underline{\mathbf{A}}$. We denote the (i, j, k)'th element by a_{ijk} (which is a scalar). We denote the L_p norm of a vector by $\|\mathbf{a}\|_p$ and the Frobenius norm of a matrix by $\|\mathbf{A}\|_F$. The $\|\mathbf{x}\|_{1/2}$ means either the L_1 norm or the L_2 norm. The mathematical definitions of L_1, L_2 and Frobenius norms are presented in Appendix A.

We denote the number of the entities of the KG by N_e, the number of the relations by N_r, the number of the observed triples by $\mathcal{D}$, the number of the observed positive triples by $\mathcal{D}^+$ and the number of the observed negative triples by $\mathcal{D}^-$.

In order to define more formally the statistical models for knowledge graph, we introduce some mathematical background. Let $\mathcal{E} = e_1, ..., e_{N_e}$ be the set of all entities and $\mathcal{R} = r_1, ..., r_{N_r}$ be the set of all relation types in a KG. All possible triples in $\mathcal{E} \times \mathcal{R} \times \mathcal{E}$ can be grouped naturally in a third-order tensor $\underline{\mathbf{Y}} \in \{0, 1\}^{N_e \times N_e \times N_r}$, *Adjacency Tensor*, where $y_{ijk} = 1$ indicates the existence

of a triple and the interpretation of $y_{ijk} = 0$ depends on whether OWA or CWA is made, i.e.:

$$y_{ijk} = \begin{cases} 1, & \text{if the triple } (e_i, r_k, e_j) \text{ exists.} \\ 0, & \text{otherwise.} \end{cases} \tag{2.1}$$

2.3 Statistical Relational Learning

Statistical Relational Learning (SRL) borrows many ideas and techniques from logic and probabilistic modeling, the main difference is its focus on learning probabilistic logical models. More specifically, it is concerned with the creation of statistical models for relational data. [9]

Except from the deterministic rules of KGs, such as type constraints and transitivity, they also have some statistical patterns or regularities. They are not universally true, but they are useful for prediction. Some of these patterns are the *homophily*, i.e. the tendency of entities to be related to other entities with similar characteristics, the *global and long-range statistical dependencies*, i.e. dependencies that can span over chains of triples and involve different types of relations, and the *block structure*, i.e. the division of entities into distinct groups (blocks), such that all the members of a group have similar relationships to members of other groups.

Nickel *et al.* [23] and Wang *et al.* [33] have made surveys of SRL methods on KGs and KG embedding. Based on these surveys, we present in this section the most common SRL models, focusing mainly on these we use in this thesis. Simple illustrations and neural networks architectures of the following models are presented in Appendix B.

2.3.1 Latent Feature Models

Semantic Matching Models. Semantic matching models exploit similarity-based scoring functions. They measure plausibility of facts by matching latent semantics of entities and relations embodied in their vector space representations.

RESCAL. RESCAL [24] explains triples via pairwise interactions of latent features. The score of a triple x_{ijk} is defined as

$$f_{ijk}^{RESCAL} := \mathbf{e}_i^\top \mathbf{W}_k \mathbf{e}_j^\top = \sum_{a=1}^{H_e} \sum_{b=1}^{H_e} w_{abk} e_{ia} e_{jb} \tag{2.2}$$

where $\mathbf{W}_k \in \mathbb{R}^{H_e \times H_e}$ is a weighted matrix associated with the k-th relation. Its space complexity is $\mathcal{O}(N_e H_e + N_r H_e^2)$ and its time complexity is $\mathcal{O}(H_e^2)$.

Matrix Factorization Methods. In matrix factorization, prior to the factorization, the adjacency tensor $\underline{\mathbf{Y}} \in \mathbb{R}^{N_e \times N_e \times N_r}$ is reshaped into a matrix $\mathbf{Y} \in \mathbb{R}^{N_e^2 \times N_r}$, by associating rows with pairs (e_i, e_j) and columns with relations r_k, [14, 15] or into a matrix $\mathbf{Y} \in \mathbb{R}^{N_e \times N_r N_r}$, by associating rows with e_i and columns with pairs (r_k, e_j) [12, 30]. However, both of these formulations lose information compared to tensor factorization. Its space complexity is $\mathcal{O}(N_e^2 H_e + N_r H_e)$, worse than the space complexity of RESCAL.

Multi-Layer Perceptrons (MLP). In MLP [7] each entity and each relation is associated with a single vector. The $\mathbf{e_i}$, $\mathbf{e_j}$ and $\mathbf{r_k}$ are concatenated in the input layer and mapped to a non-linear hidden layer. The score is generated by a linear output layer, i.e.

$$f_{ijk}^{MLP} := \mathbf{w}^\top \tanh(\mathbf{M}_1\mathbf{e_i} + \mathbf{M}_2\mathbf{e_j} + \mathbf{M}_3\mathbf{r_k}) \tag{2.3}$$

where $\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3 \in \mathbb{R}^{H_e \times H_e}$ are the first layer weights and $\mathbf{w} \in \mathbb{R}^{H_e}$ the second layer weights. MLPs can learn to put "semantically similar" words close by in the embedding space, even if they are not explicitly trained to do so. [22] Its space complexity is $\mathcal{O}(N_e H_e + N_r H_e)$ and its time complexity is $\mathcal{O}(H_e^2)$.

Neural Tensor Networks (NTN). NTN [29] is a combination of RESCAL and MLP models. Given a fact, it first projects entities to their vector embeddings in the input layer. Then, the two entities $e_i, e_j \in \mathbb{R}^{H_e}$ are combined by a relation-specific tensor $\underline{\mathbf{M}}_r \in \mathbb{R}^{H_e \times H_e \times H_r}$(along with other parameters) and mapped to a non-linear hidden layer. Finally, a relation-specific linear output layer gives the score

$$f_{ijk}^{NTN} := \mathbf{w}^\top \tanh(\mathbf{e_i}^\top \underline{\mathbf{M}}_r \mathbf{e_j} + \mathbf{M}_r^1\mathbf{e_i} + \mathbf{M}_r^2\mathbf{e_j} + \mathbf{b}_r) \tag{2.4}$$

where $\mathbf{M}_r^1, \mathbf{M}_r^2 \in \mathbb{R}^{H_e \times H_r}$ are relation-specific weight matrices and $\mathbf{b}_r \in \mathbb{R}^{H_r}$ is a bias vector. NTN might be the most expressive model to date, but it is not efficient in handling large-scale KGs, because its space complexity is $\mathcal{O}(N_e H_e + N_r H_e^2 H_r)$ and its time complexity is $\mathcal{O}(H_e^2 H_r)$.

Translational Distance Models. Translational distance models exploit distance-based scoring functions. They measure the plausibility of a fact as the distance between the two entities, usually after a translation carried out by the relation.

TransE. TransE [2] is an energy based model for learning low-dimensional embeddings of entities. Relations are represented as *translations in the embedding space*, such that given a fact (h, r, t), the embedding of the tail t should be close to the embedding of the head h plus some vector that depends on the relation r. In other words, $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ when (h, r, t) holds, considering the translation vector $\mathbf{r}$ and the embedded entities $\mathbf{h}$ and $\mathbf{t}$. The scoring function of a triple x_{ijk} is defined as the (negative) distance between $\mathbf{h} + \mathbf{r}$ and $\mathbf{t}$, i.e.

$$f_{ijk}^{TransE} := -\|\mathbf{e_i} + \mathbf{r_k} - \mathbf{e_j}\|_{1/2} \tag{2.5}$$

The score is expected to be large, if the triple x_{ijk} holds.

The basic idea behind TransE is that if (h, r, t) holds, then $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ and $\mathbf{t}$ should be the nearest neighbor of $\mathbf{h} + \mathbf{r}$, while $\mathbf{h} + \mathbf{r}$ should be far away from $\mathbf{t}$ otherwise. To learn such embeddings, a margin-based ranking criterion is minimized over the training set:

$$\mathcal{L} = \sum_{(h,r,t)\in\mathcal{D}^+} \sum_{(h',r,t')\in\mathcal{D}'_{(h,r,t)}} [\gamma + \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_{1/2} - \|\mathbf{h}' + \mathbf{r} - \mathbf{t}'\|_{1/2}]_+ \tag{2.6}$$

Algorithm 1: Learning TransE

Input: Training set $\mathcal{D}^+$, entities set $\mathcal{E}$, relations set $\mathcal{R}$, margin γ, embeddings dimension k

initialize $\mathbf{r} \leftarrow$uniform$(-\frac{6}{\sqrt{k}}, \frac{6}{\sqrt{k}})$ for each $r \in \mathcal{R}$
  $\mathbf{r} \leftarrow \mathbf{r}/\|\mathbf{r}\|$ for each $r \in \mathcal{R}$
  $\mathbf{e} \leftarrow$uniform$(-\frac{6}{\sqrt{k}}, \frac{6}{\sqrt{k}})$ for each $e \in \mathcal{E}$
loop
  $\mathbf{e} \leftarrow \mathbf{e}/\|\mathbf{e}\|$ for each $e \in \mathcal{E}$
  $\mathcal{D}_{batch} \leftarrow$ sample $(\mathcal{D}^+, b)$ // sample a minibatch of size b
  $\mathcal{T}_{batch} \leftarrow \emptyset$ // initialize the set of pairs of triples
  for $(h, r, t) \in \mathcal{D}_{batch}$ **do**
    $(h', r, t') \leftarrow$ sample$(\mathcal{D}'_{(h,r,t)})$ // sample a corrupted triple
    $\mathcal{T}_{batch} \leftarrow \mathcal{T}_{batch} \cup \{((h, r, t), (h', r, t'))\}$
  end
  Update embeddings w.r.t.
  $\sum_{((h,r,t),(h',r,t'))\in\mathcal{T}_{batch}} \nabla[\gamma + \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_{1/2} - \|\mathbf{h}' + \mathbf{r} - \mathbf{t}'\|_{1/2}]_+$
end

where $[x]_+$ denotes the positive part of x, $\gamma > 0$ is a margin hyperparameter and $\mathcal{D}'_{(h,r,t)}$ is the set of corrupted triples given the fact (h, r, t). The $\mathcal{D}'_{(h,r,t)}$ set is defined as:

$$\begin{aligned}\mathcal{D}'_{(h,r,t)} = &\{(h', r, t) \mid h \neq h' \wedge (h, r, t) \in \mathcal{D}^+ \wedge h' \in \mathcal{E}\} \\ &\cup \{(h, r, t') \mid t \neq t' \wedge (h, r, t) \in \mathcal{D}^+ \wedge t' \in \mathcal{E}\}\end{aligned} \tag{2.7}$$

The optimization is carried out by stochastic gradient descent. The detailed optimization by Bordes *et al.* [2] is described in Algorithm 1.

Although the TransE model is simple and efficient, it has flaws in dealing with 1-to-N, N-to-1 and N-to-N relations. For example in 1-to-N relations, given a relation r and an entity h, TransE enforces $\mathbf{h} + \mathbf{r} \approx \mathbf{t_i}$ for all $i = 1, ..., p$, such that $(h, r, t_i) \in \mathcal{D}^+$, and then $t_1 \approx ... \approx t_p$. Similar disadvantages exist for N-to-1 and N-to-N relations.

Its space complexity is $\mathcal{O}(N_e H_e + N_r H_r)$ and its time complexity is $\mathcal{O}(H_e)$.

Extensions of TransE. In order to overcome the disadvantages of TransE in dealing with 1-to-N, N-to-1 and N-to-N relations, the TransE model has been extended by allowing an entity to have distinct representations when involved in different relations. Some of its extensions are the following:

- TransH [34], which introduces relation-specific hyperplanes. Entities are modeled again as vectors, but each relation r as a vector $\mathbf{r}$ on a hyperplane.
- TransR [19], which introduces relation-specific spaces. Entities are represented as vectors in an entity space $\mathbb{R}^{H_e}$ and each relation is associated with a specific space $\mathbb{R}^{H_r}$ and modeled as a translation vector in that space. In addition, TransR introduces a projection matrix for each relation, which requires $\mathcal{O}(H_e H_r)$ parameters per relation, so it loses the simplicity and efficiency of TransE.
- TransD [13], which decomposes further the projection matrix into a product of two vectors.

PTransE. TransE and its extensions mentioned above take into consideration only direct relations between entities. Path-based TransE (PTransE) [18] builds also triples of KBs using entity pairs connection with relation paths. Given a path $p = (r_1, ..., r_l)$ linking two entities h and t, PTransE considers three types of composition operations:

- Addition (ADD): $\mathbf{p} = \mathbf{r}_1 + ... + \mathbf{r}_l$
- Multiplication (MUL): $\mathbf{p} = \mathbf{r}_1 \circ ... \circ \mathbf{r}_l$
- Recurrent Neural Network (RNN): $\mathbf{c}_i = f(\mathbf{W}[\mathbf{c}_{i-1}; \mathbf{r}_i])$, where $\mathbf{c}_i$ indicates the accumulated path vector at the i-th relation.

As not all relation paths are meaningful and reliable for learning, PTransE computes the path reliability of each path, based on path-constraint resource allocation (PCRA). Given a path triple (h, p, t), the flowing path can be written as $S_0 \xrightarrow{r_1} S_1 \xrightarrow{r_2} ... \xrightarrow{r_l} S_l$, where $S_0 = h$ and $s_l = t$. For each entity $m \in S_i$, $S_{i-1}(\cdot, m)$ is the direct predecessors along relation r_i in S_{i-1}. The reliability of path p given (h, m) is defined as

$$R(p \mid h, m) = R_p(m) = \sum_{n \in S_{i-1}(\cdot, m)} \frac{1}{\mid S_i(n, \cdot) \mid} R_p(n) \tag{2.8}$$

where $S_i(n, \cdot)$ is the direct successors of $n \in S_{i-1}$ following the relation r_i. For each relation path p, the head h has $R_p(h) = 1$.

For each fact $(h, r, t) \in \mathcal{D}^+$, PTransE defines a loss w.r.t. the paths as

$$\mathcal{L}_{path} = \frac{1}{Z} \sum_{p \in P(h,t)} R(p \mid h, t) \cdot E(h, p, t) \tag{2.9}$$

where $P(h, t)$ is the set of all paths between h and t, $Z = \sum_{p \in P(h,t)} R(p \mid h, t)$ is a normalization factor and the energy function $E(h, p, t)$ is defined as

$$E(h, p, t) = \|\mathbf{h} + \mathbf{p} - \mathbf{t}\| = \|\mathbf{p} - \mathbf{r}\| \tag{2.10}$$

PtransE adds also the reverse relations for each relation in KBs, i.e. a triple (t, r^{-1}, h) is built for each $(h, r, t) \in \mathcal{D}^+$.

Its space complexity is $\mathcal{O}(N_e H_e + N_r H_r)$ and its time complexity is $\mathcal{O}(H_e P L)$ for ADD and MUL and $\mathcal{O}(H_e^2 P L)$ for RNN, where P is the number of relation paths between two entities and L is the length of relation paths.

2.3.2 Graph Feature Models

Graph Feature Models' approaches assume that the existence of an edge can be predicted by extracting features from the observed edges in the graph. In contrast to latent feature models, this kind of reasoning explains triples directly from the observed data in KG. These methods are based on homophily, while similarity of entities can be derived from the neighborhood of nodes or from the existence of paths between nodes. [23] We assume that all y_{ijk} are conditionally independent

given observed graph features and additional parameters. The existence of a triple x_{ijk} is predicted by a score function, which represents the model's confidence that the triple exists.

The Path Ranking Algorithm (PRA) [16, 17] uses random walks of bounded length on the graph and compute the probability of each path. The main idea of PRA is to use these path probabilities as supervised features for each entity pair and use any favorable classification model, such as logistic regression and SVM, to predict the probability of missing edges. [35] In particular, given a direct edge from e_i to e_j, let $\pi_L(i, j, k, t)$ denote a path of length L, where t represents the sequence of edge types $t = \{r_1, r_2, ..., r_L\}$. Let $\Pi_L(i, j, k)$ represent the set of all such paths of length L and $P(\pi_L(i, j, k, t))$ the probability of this particular path.

2.3.3 Markov Random Fields

A Markov network (also known as Markov Random Field) is a model for the joint distribution of a set of variables $X = (X_1, X_2, ..., X_n) \in X$ [26]. Markov Logic Networks (MLNs) are a SRL representation that instantiates to an undirected graphical model. Graphical models use graphs to encode dependencies between random variables. This fact is also their main difference from KGs, which encode the existence of facts. Each possible fact y_{ijk} (or random variable), which can depend on any other $N_e \times N_e \times N_r - 1$ random variables in $\mathbf{Y}$, is represented as a node in the graph, while each dependency between them is represented as an edge. These graphs are called *dependency graphs*. [23]

Each relational feature in a Markov Logic Network (MLN) is specified as first-order logic rule R_ϕ with an attached weight λ_ϕ. MLNs can be viewed as relational analogs to Markov networks, in which the potential functions over cliques are defined by the instantiations of the formulae in $\mathcal{F}$. The probability of a possible world $\mathbf{x}$ is thus given by:

$$P(\mathbf{X} = \mathbf{x}) = \frac{exp(\sum_{R_\phi \in \mathcal{F}} \lambda_\phi n_\phi(\mathbf{x}))}{Z} \tag{2.11}$$

where $n_\phi(\mathbf{x})$ is the number of true instantiations of rule R_ϕ and Z is a normalizing constant. [9]

To explain this further, consider an example by Domingos *et al.* [27] about the patterns of human interactions and smoking habits. According to the example, friends tend to have similar smoking habits, i.e. if two people are friends, either both smoke or neither does. This fact can be captured with the following first-order logic rule (where λ is the weight associated with it):

$$\lambda : \forall x \forall y Friends(x, y) \Rightarrow (Smokes(x) \Leftrightarrow Smokes(y)). \tag{2.12}$$

Figure 2.1 shows the graph of the ground Markov network defined by this formula.

2.4 Negative Sampling

A common problem is that most KGs contain only positive training examples, i.e. $y_{ijk} = 1$ for all $(i, j, k) \in \mathcal{D}$. Training on all-positive data could be catastrophic, because the model might easily over generalize. Hence, there is the need of negative sampling, the creation of false triples. [8] Different approaches to create negative examples have been proposed [23], such as:

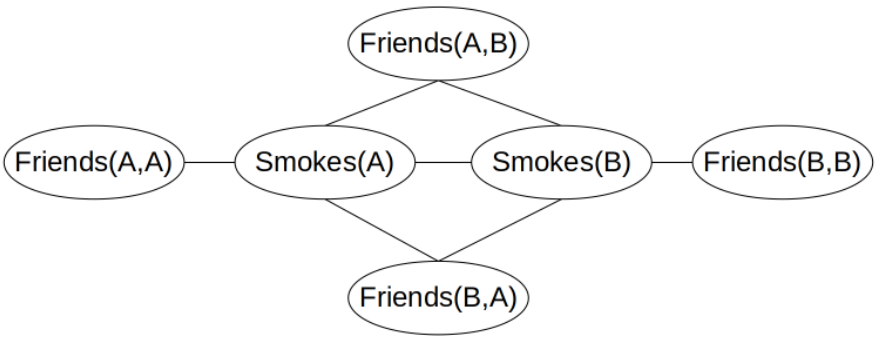


Figure 2.1: Ground Markov network obtained by applying the Folmula 2.12 [27]

- to make CWA and assume that all (type consistent) triples that are not in $\mathcal{D}$ are false. For incomplete KGs this assumption will be violated and lead to scalability issues.

- to exploit type, functional or valid value ranges' constraints on the structure of a KG. However, functional constraints are scarce and negative examples based on type constraints and valid value ranges are usually not sufficient for training.

- to corrupt true triples by replacing either the head or the tail with a random entity from $\mathcal{E}$. In particular,

$$\begin{aligned}\mathcal{D}^- = &\{(e_l, r_k, e_j) \mid e_i \neq e_l \wedge (e_i, r_k, e_j) \in \mathcal{D}^+\} \\ &\cup \{(e_i, r_k, e_l) \mid e_j \neq e_l \wedge (e_i, r_k, e_j) \in \mathcal{D}^+\}\end{aligned} \tag{2.13}$$

- to corrupt true triples by replacing either the head or the tail with a random entity from $\mathcal{E}$ or the relation with a random one from $\mathcal{R}$. In particular,

$$\begin{aligned}\mathcal{D}^- = &\{(e_l, r_k, e_j) \mid e_i \neq e_l \wedge (e_i, r_k, e_j) \in \mathcal{D}^+\} \\ &\cup \{(e_i, r_k, e_l) \mid e_j \neq e_l \wedge (e_i, r_k, e_j) \in \mathcal{D}^+\} \\ &\cup \{(e_i, r_l, e_j) \mid r_k \neq r_l \wedge (e_i, r_k, e_j) \in \mathcal{D}^+\}\end{aligned} \tag{2.14}$$

- to make a *Local-Closed World Assumption* (LCWA). We assume that a KG is only *locally* complete. More precisely, if there is any triple for a particular subject-predicate pair (e_i, r_k), then any non-existing triple $(e_i, r_k, \cdot)$ is indeed false and included them in $\mathcal{D}^-$. The assumption is made only for functional relations. If there is no observed triple for the pair (e_i, r_k), all triples $(e_i, r_k, \cdot)$ are considered as unknown and not included in $\mathcal{D}^-$.

2.5 Summary of Useful Definitions

In this section some definitions of concepts used in the text are described. [5, 23]

- *Knowledge Graph* models information in the form of entities and relationships between them. Nodes represent entities and edges represent existing relationships.

- *Statistical Relational Learning* is concerned with the creation of statistical models for relational data.
- *Link Prediction/Knowledge Graph Prediction* is concerned with predicting the existence (or probability of correctness) of edges in the graph.
- *Embedding* is a mapping from discrete objects, such as words, to vectors of real numbers.
- *Closed World Assumption (CWA)*, non-existing triples indicate false relationships.
- *Open World Assumption (OWA)*, a non-existing triple is interpreted as unknown, i.e. the corresponding relationship can be either true or false. RDF makes the OWA.
- *Adjacency Tensor* is a third-order tensor $\underline{\mathbf{Y}} \in \{0,1\}^{N_e \times N_e \times N_r}$, where $y_{ijk} = 1$ indicates the existence of a triple and the interpretation of $y_{ijk} = 0$ depends on whether OWA or CWA is made.
- *Negative Samples/Examples* are the observed false facts (triples) in the dataset.
- *Latent Features* are the features, which are not directly observed in the data.

2.6 Summary of the Notation

The Table 2.1 describes summarily some of the notation mentioned above and used in the following chapters.

Table 2.1: Summary of the notation [18, 23]

Relational Data		
Symbol	Meaning	
N_e	Number of entities	
N_r	Number of relations	
N_d	Number of training examples	
e_i	i-th entity in the dataset	
r_k	k-th relation in the dataset	
D^+	Set of observed positive triples	
D^-	Set of observed negative triples	
Latent Feature Models		
Symbol	Meaning	Size
H_e	Number of latent features for entities	
H_r	Number of latent features of relations	
e_i	Latent feature of the entity e_i	H_e
r_k	Latent feature of the relation r_k	H_r
Other Symbols		
Symbol	Meaning	Size
$\underline{\mathbf{Y}}$	Adjacency tensor	$N_e \times N_e \times N_r$
$R(p\|h,t)$	Reliability of path p given (h,t)	

Chapter 3

Framework

3.1 Introduction

As mentioned in Section 2.3.1, although TransE and its extensions are efficient, simple and successful in modeling relational facts, they take into consideration only direct relations between entities. However, there are also substantial multiple-step relation paths between entities indicating their semantic relationships. For example, the relation path $h \xrightarrow{\text{BornInCity}} e_1 \xrightarrow{\text{CityInCountry}} t$ indicates the relation `Nationality` between h and t, i.e. $(h, \texttt{Nationality}, t)$.

On the other hand, PTransE adds in KBs not only direct relations, but also multiple-step relation paths and reverse relations. In some cases, the reverse versions of relations may be useful, but usually they are not presented in KBs, as shown in Table 3.1. For instance, the fact $h \xrightarrow{hasDirector} t$ indicates the relation $t \xrightarrow{hasDirector^{-1}} h$ or $t \xrightarrow{isDirectorOf} h$. The main disadvantage of PTransE is that it has flaws in time complexity and it impairs the efficiency of TransE.

Table 3.1: Difference between TransE and PTranse [18]

	TransE	PTransE	
KB	$h \xrightarrow{r} t$	$h \xrightarrow{r_1} e_1 \xrightarrow{r_2} t$	$t \xrightarrow{r^{-1}} h$
Triples	(h, r, t)	$(h, r_1, e_1)\ (e_1, r_2, t)$ $(h, r_1 \circ r_2, t)$	(t, r^{-1}, h)
Objective	$\mathbf{h} + \mathbf{r} = \mathbf{t}$	$\mathbf{h} + (\mathbf{r_1} \circ \mathbf{r_2}) = \mathbf{t}$	$\mathbf{t} + \mathbf{r}^{-1} = \mathbf{h}$

3.2 Our Model

In this thesis, we introduce the TransEP model, which is a combination of TransE and PTransE. We aim to maintain the efficiency of TransE and take advantage of the PTransE's extra knowledge.

Before training, as shown in Figure 3.1, we add in the KG the multiple-step paths and the reverse relations and calculate the reliability of each path, see Figure 3.2. Furthermore, as it will be impractical to enumerate all possible relation paths between entities without limitation, we remove triples with very low reliability score.

Since different triples have different contribution to the global information content of the KG, the loss function should consider a triple more if it has larger information content. Thus, after obtaining the triple weights/reliability, we modify the TransE's margin-based loss function and deploy a weighted-margin-based loss function, such as Mai *et. al* in [20].

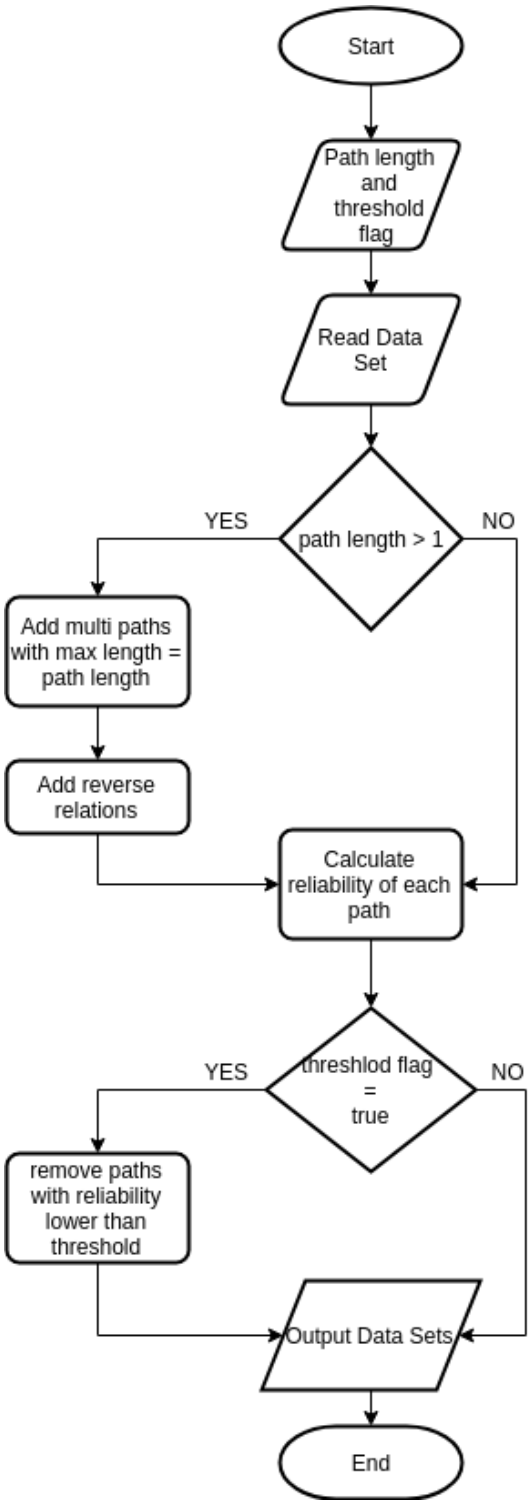


Figure 3.1: Flowchart of Data Set Creation

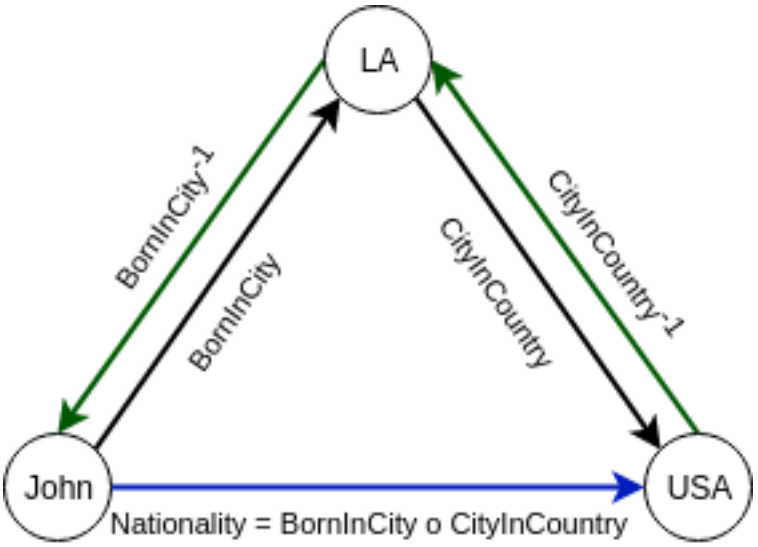


Figure 3.2: Example of KG after adding multiple-step relation paths (blue arrow) and reverse relations (green arrows).

Chapter 4

Methodology

In this thesis we study data extracted from Freebase and SNOMED CT. Their statistics are given in Table 4.1.

4.1 Data Sets

4.1.1 FB15K

Freebase contains currently around 1.9 billion triples and hundreds or thousands of relation types pertaining to world-knowledge obtained autmoatically or semi-automatically from various resources on millions of entities. These relations include born-in, nationality, is-in (for geographical entities), part-of (for organizations, among others), and more. The FB15K dataset was introduced by Bordes *et al.* in [2]. It is a subset of Freebase, which contains about 14,951 entities with 1,345 different relations.

4.1.2 SNOMED CT

SNOMED CT (Systematized Nomenclature of Medicine – Clinical Terms) is a systematically organized computer processable collection of medical terms providing codes, terms, synonyms and definitions used in clinical documentation and reporting. It encodes the meanings that are used in health information and supports the effective clinical recording of data with the aim of improving patient care. SNOMED CT comprehensive coverage includes: clinical findings, symptoms, diagnoses, procedures, body structures, organisms and other etiologies, substances, pharmaceuticals, devices and specimens. [1]

The International Health Terminology Standards Development Organization (IHTSDO) describes in [25] the basics, the concepts and the properties of SNOMED CT. Its content is represented using three types of components:

- *Concepts* representing clinical meanings that are organized into hierarchies.
- *Descriptions*, which link appropriate human readable terms to concepts.
- *Relationships*, which link each concept to other related concepts.
- Components are supplemented by *reference sets*, which provide additional flexible features and enable configuration of the terminology to address different requirements.

Furthermore, concepts are related to one another by relationships and Web Ontology Language (OWL) [21] axioms that provide a formal logical definition of the concept. In other words, SNOMED CT contains a $\mathcal{TBOX}$ containing a set of axioms. Transitivity, see Figure 4.1, and property chains, which are are in some ways similar to transitivity but involve more than one attribute, see figure 4.2, are some common axioms and can enhance classification.

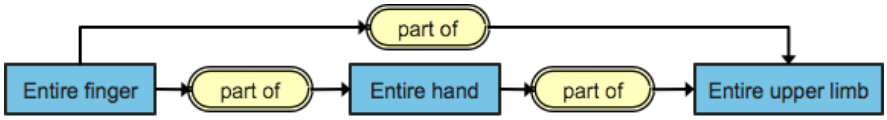


Figure 4.1: Example of SNOMED CT Transitivity Property [25]

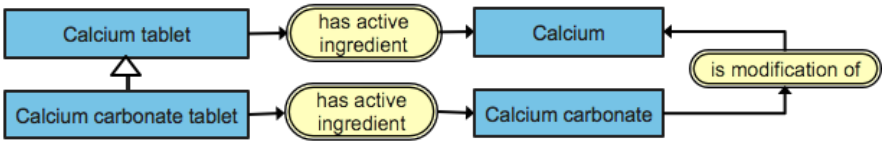


Figure 4.2: Example of SNOMED CT Property Chain [25]

SNOMED CT concepts are organized in hierarchies. Within a hierarchy concepts range from the more general to the more detailed. Related concepts in the hierarchy are linked using the |is a| relationship. Support for multiple levels of granularity allows SNOMED CT to be used to represent clinical data at a level of detail that is appropriate to a range of different uses. Concepts with different levels of granularity are linked to one another by |is a| relationships, see Figure 4.3. This enables appropriate aggregation of specific information within less detailed categories.

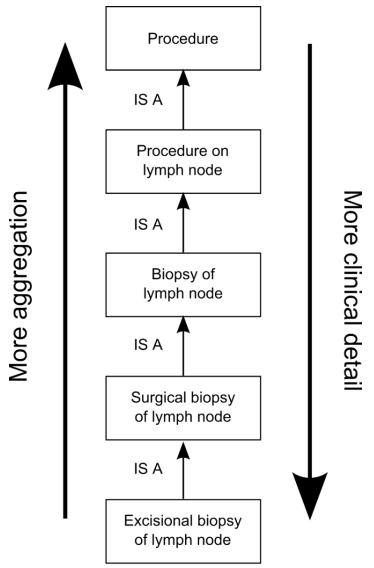


Figure 4.3: Example of SNOMED CT Multiple levels of granularity [25]

Construction of Data Set After downloading the SNOMED CT International Edition RF2 (Release Format 2) from the U.S. National Library of Medicine[1], we used the GitHub repository by Rory Davidson [6] to construct the Neo4j Graph Database, see a part of the graph in Figure 4.4.

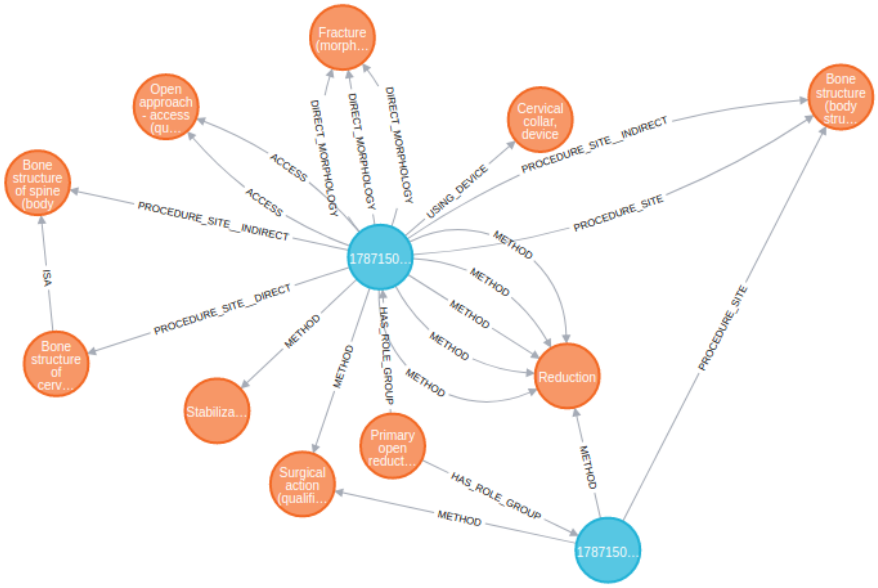


Figure 4.4: Example of SNOMED CT Neo4j Graph Database

In order to construct the data, we used the following queries in Cypher Query Language[2]:

- `MATCH (x:ObjectConcept) RETURN x.FSN, n.sctid`, which returns the name and the id of each entity
- `MATCH ()-[r]-() RETURN type(r), r.typeId`, which returns the type and the id of each relation
- `MATCH (x:ObjectConcept) MATCH (x)-[r:ISA]->(y:ObjectConcept) RETURN x.sctid, r.typeId, y.sctid`, which returns the entities' ids and the ISA relation's type id of each (x, ISA, y) triple
- `MATCH (x:ObjectConcept) MATCH (x)-[:HAS_ROLE_GROUP]->(rg:RoleGroup) MATCH (rg:RoleGroup)-[r]->(y:ObjectConcept) RETURN DISTINCT x.sctid, y.sctid, r.typeId`, which returns the entities' ids and the relation's type id of each (x, r, y) triple

[1] https://www.nlm.nih.gov/healthit/snomedct/
[2] https://neo4j.com/developer/cypher/

Table 4.1: Statistics of Data Sets

	FB15K	SNOMED
Number of Entities	$14,951$	$466,612$
Number of Relationships	$1,345$	113
Number of Train Data	$483,142$	$1,430,419$
Number of Valid Data	$50,000$	$357,605$
Number of Test Data	$59,071$	$447,007$

4.2 Construction of reverse and multiple-step Paths

4.2.1 Reverse Relations

We add in KGs of both data sets the reverse relations for each observed fact, i.e. for each triple (h, r, t) we build another (t, r^{-1}, h).

4.2.2 Multiple-step Paths

FB15K As FB15K contains thousands of relation types pertaining to world-knowledge without any particular subject, we consider multiple-step relation paths between two entities e_1 and e_2, only if there is at least one observed fact (e_1, r, e_2), where $r \in \mathcal{R}$. We construct the multiple-step relation paths according to Algorithm 2.

Algorithm 2: Construction of multiple-step relation paths on FB15K

Input: Training set $\mathcal{D}^+$, entities set $\mathcal{E}$, relations set $\mathcal{R}$, max length l
initialize $\mathcal{P} \leftarrow \mathcal{D}^+$ // $\mathcal{P}$ is the set of triples with length = currentLength - 1
$\mathcal{N} \leftarrow \emptyset$ // initialize the set of new triples
for $i = 2; i <= l; i++$ **do**
  $\mathcal{N}_i \leftarrow \emptyset$ // initialize the set of new triples of length i
  for $(e_1, r_1, e_2) \in \mathcal{P}$ **do**
    for $(e_2, r_2, e_3) \in \mathcal{D}^+$ **do**
      if $\exists (e_1, r, e_3) \in \mathcal{D}^+$ **then**
        Add $(e_1, r_1 \circ r_2, e_3)$ in $\mathcal{N}$ and $\mathcal{N}_i$
      end
    end
  end
  $\mathcal{P} \leftarrow \mathcal{N}_i$
end

SNOMED CT SNOMED CT contains only a few relation types, but a lot of OWL axioms. This indicates the existence of relations between two entities e_1 and e_2 without any observed fact (e_1, r, e_2). Thus, we consider multiple-step relation paths between all entities pairs. In addition, we take under consideration the transitivity property of relation types ISA and $PART\_OF$. The construction of multiple-step relation paths is described in Algorithm 3.

Algorithm 3: Construction of multiple-step relation paths on SNOMED CT

Input: Training set $\mathcal{D}^+$, entities set $\mathcal{E}$, relations set $\mathcal{R}$, max length l

initialize $\mathcal{P} \leftarrow \mathcal{D}^+$ // $\mathcal{P}$ is the set of triples with length = currentLength - 1
$\mathcal{N} \leftarrow \emptyset$ // initialize the set of new triples
for $i = 2; i <= l; i++$ **do**
  $\mathcal{N}_i \leftarrow \emptyset$ // initialize the set of new triples of length i
  for $(e_1, r_1, e_2) \in \mathcal{P}$ **do**
    for $(e_2, r_2, e_3) \in \mathcal{D}^+$ **do**
      if $r_2 == PART\_OF$ **then**
        $r \leftarrow r_1$
      else if $r_1 == ISA$ **then**
        $r \leftarrow r_2$
      else if $r_2 == ISA$ **then**
        $r \leftarrow r_1$
      else
        $r \leftarrow r_1 \circ r_2$
      end
      Add (e_1, r, e_3) in $\mathcal{N}$ and $\mathcal{N}_i$
    end
  end
  $\mathcal{P} \leftarrow \mathcal{N}_i$
end

4.3 Reliability of Paths

As mentioned in PTransE's description, not all relation paths are meaningful and reliable for learning. Thus, we calculate the reliability score of each path by using the following equation

$$R(p \mid h, m) = R_p(m) = \sum_{n \in S_{i-1}(\cdot, m)} \frac{1}{\mid S_i(n, \cdot) \mid} R_p(n), \tag{4.1}$$

which is also used by PTransE and described thoroughly in Subparagraph 13. After calculating all scores, we divide them by the normalization factor $Z = \sum_{p \in P(h,t)} R(p \mid h, t)$ of PTransE. Therefore, the range of the reliability score is $(0, 1]$.

We obtain the reliability of each path before training, as shown in Figure 3.1, in order not to impair the time complexity of TransE model.

In KGs containing only direct relations we calculate the reliability score according to Equation 4.1. On the other hand, for training on KGs containing also reverse relations and multiple-step relation paths, we consider the reliability score of direct paths as 1.0, while the score of all the other paths according to Equation 4.1, because the observed facts are more likely to be meaningful for learning.

There are usually large amount of relations and facts about each entity pair and it will be impractical to enumerate all possible relation paths between head and tail entities. For computational efficiency, in some cases we consider only these relation paths with the reliability score larger than 0.01.

4.4 Loss Function

We denote the reliability score of the path $h \xrightarrow{r} t$ by $w_{(h,r,t)}$. To learn the KG embedding, we use a margin based loss function, as TransE does. However, according to the idea of Mai *et. al* in [20], in the loss function we multiply $w_{(h,r,t)}$ with the subtraction value between the scoring function, see Eq. 2.5, of the triple (h, r, t) and the scoring function of the corrupted triple (h', r, t'):

$$\mathcal{L} = \sum_{(h,r,t)\in\mathcal{D}^+} \sum_{(h',r,t')\in\mathcal{D}'_{(h,r,t)}} [\gamma + w_{(h,r,t)} \cdot (\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_{1/2} - \|\mathbf{h}' + \mathbf{r} - \mathbf{t}'\|_{1/2})]_+ \tag{4.2}$$

As mentioned in Section 3.2, the loss function should consider a triple more if it has larger information content.

4.5 Negative Sampling

For negative sampling, we follow the idea described in Equation 2.13, where we replace randomly either the head or the tail with a random entity from $\mathcal{E}$ to construct negative triples.

Chapter 5

Implementation

Our approach, TrasnEP, is evaluated on data extracted from Freebase and SNOMED CT described thoroughly in subsections 4.1.1 and 4.1.2 respectively.

For our experiments, we use the GitHub repository [11] by Han *et. al* [10] implemented with PyTorch. We used their TransE implementation and modified it in order to implement TransEP.

5.1 Evaluation Protocol

For evaluation, the same ranking procedure as in [3] is used. For each test triple, the head is removed and replaced by each of the entities in $\mathcal{E}$. Dissimilarities (or energies) of those corrupted triplets are first computed by the models and then sorted by ascending order; the rank of the correct entity is finally stored. This whole procedure is repeated while removing the tail instead of the head.

5.1.1 Metrics

Aggregating over all the N_{test} testing triples, we use four metrics to do the evaluation:

Mean Rank (MR) The value of averaged rank or Mean Rank. The smaller, the better. MR is calculated by:

$$MR = \frac{1}{2N_{test}} \sum_{i=1}^{N_{test}} (rank_{ih} + rank_{it}) \tag{5.1}$$

where $rank_{ih}$ and $rank_{ih}$ refer to the rank position of the prediction of the i_{th} correct triple by corrupting the head or the tail respectively.

Mean Reciprocal Rank (MRR) The mean of all reciprocal ranks for the true candidates over the test set. The higher, the better. MRR is calculated by:

$$MRR = \frac{1}{2N_{test}} \sum_{i=1}^{N_{test}} (\frac{1}{rank_{ih}} + \frac{1}{rank_{it}}) \tag{5.2}$$

where $rank_{ih}$ and $rank_{ih}$ have the same meaning as in MR. [32]

Hits at k (Hits@k) The rate of correct entities appearing in the top k entries for each instance list. This number may exceed 1.00 if the average k-truncated list contains more than one true entity. The

higher, the better. As described by Bordes *et. al* [2], Hits@k is calculated by:

$$Hits@k = \frac{1}{2N_{test}} \sum_{i=1}^{N_{test}} (I_k(rank_{ih}) + I_k(rank_{it})) \tag{5.3}$$

$$I_k(rank_i) = \begin{cases} 1, & \text{if } rank_{\leq} k, \\ 0, & \text{otherwise.} \end{cases} \tag{5.4}$$

Triple Classification Accuracy The rate of the correct predictions made in test set. Accuracy is calculated by:

$$Accuracy = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \tag{5.5}$$

Based on the valid set we calculate a threshold for the triples' scores. True samples have a score lower than the threshold, while negative ones have a score greater than the threshold.

5.1.2 Filtering

The metrics described above are indicative, but may have flaws when some corrupted triples end up being true, from the training set for instance. In this case, those may be ranked above the test triple, but this should not be counted as an error, as both triples are true. To avoid such a misleading behavior, Bordes *et al.* in [2] proposed to remove from the list of corrupted triples all the triples that appear either in the training, validation or test set (except the test triple of interest). This ensures that all corrupted triples do not belong to the data set. In the following, mean ranks and hits@k are reported according to both settings: the original one is termed *raw*, while we refer to the newer as *filter*.

5.1.3 Type Constraints

Han *et. al* in [10] creates a type constraining file, which contains type constraints for each relation, i.e. which entities every relation has as head entities and as tail ones. According to this file, for each test triple, the head or the tail is removed and replaced only by entities contained in the type constraints of each relation. In the following, mean ranks and hits@k are also reported according to type constraints settings.

5.2 Experimental Setup

For experiments with TransE and TranEP, we selected the learning rate λ for the stochastic gradient descent among $\{0.001, 0.01, 0.1\}$, the margin γ among $\{1, 2\}$ and the latent dimension k among $\{50, 100\}$ on the validation set of each data set. The dissimilarity measure d was set to the L_1. Optimal configurations for both data sets, FB15K and SNOMED CT, were: $k = 100$, $\lambda = 0.001$, $\gamma = 1$, and $d = L_1$. For both data sets, training time was limited to at most $1,000$ epochs over the training set. For FB15K, the validation is applied every 10 epochs, while for SNOMED CT every 100 epochs, due to the large number of entities. The best models were selected by early stopping using the mean predicted ranks on the validation sets (*raw* setting).

5.3 Experiments

We implemented experiments with TransE and TransEP on FB15K and SNOMED CT. For experiments on both data sets, we trained both models on three KGs: (1) containing only direct relations, (2) containing direct and reverse relations and multiple-step relation paths of length 2 and (3) containing direct and reverse relations and multiple-step relation paths of length 2 and 3 and removing triples with reliability score lower than threshold. As in FB15K there were many facts with very low reliability score, we also trained the models on a fourth KG (4) which is like the second one, but without triples with reliability score lower than threshold. We restrict the length of the paths at most 3-steps, not only for computational efficiency, but also because there was no improvement in results.

Chapter 6

Evaluation

In this chapter we will use the parameters chosen in Section 5.2 in order to implement the experiments described in Section 5.3. In the following tables we report the performance of TransE and TransEP (1) with only direct relations, (2) with reverse relations and multiple-step relation paths (step-k) and (3) without triples with reliability score lower than threshold (step-k-thres.).

6.1 Results on FB15K

Tables 6.1, 6.2 and 6.3 display the results on FB15K for all compared methods, see Section 5.1.

Table 6.1: FB15K no type constraint results

Method	MRR(%)		MR		Hits@10(%)		Hits@3(%)		Hits@1(%)	
	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter
TransE	**24.79**	**25.8**	**219**	**203**	**50.33**	**51.97**	**28.63**	**30.14**	**12.9**	**13.54**
TransE (step-2)	18.26	19.0	234	217	48.8	50.37	23.77	25.16	3.86	4.1
TransE (step-2-thres.)	18.9	19.7	228	211	48.64	50.23	24.24	25.63	4.78	5.1
TransE (step-3-thres.)	14.4	14.95	242	226	42.83	44.36	17.87	18.91	1.45	1.51
TransEP	23.61	24.5	244	228	48.43	49.95	26.95	28.4	12.09	12.63
TransEP (step-2)	18.37	19.07	254	238	46.28	47.78	22.63	23.84	5.31	5.58
TransEP (step-2-thres.)	19.79	20.6	245	229	47.56	49.07	24.17	25.48	6.79	7.19
TransEP (step-3-thres.)	15.41	15.99	254	237	41.98	43.39	18.47	19.49	3.29	3.47

Table 6.2: FB15K type constraint results

Method	MRR(%)		MR		Hits@10(%)		Hits@3(%)		Hits@1(%)	
	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter
TransE	26.61	27.68	**180**	**163**	51.8	53.5	30.21	31.77	14.97	15.68
TransE (step-2)	**27.53**	**28.57**	185	169	**52.52**	**54.12**	**31.1**	**32.65**	**16.05**	**16.75**
TransE (step-2-thres.)	26.64	27.71	184	168	51.79	53.44	30.18	31.74	15.02	15.77
TransE (step-3-thres.)	25.31	26.29	187	171	49.44	51.02	28.49	29.94	14.12	14.78
TransEP	25.73	26.67	189	173	50.34	51.92	28.94	30.44	14.4	15.02
TransEP (step-2)	26.56	27.51	193	177	50.84	52.37	29.59	30.98	15.45	16.09
TransEP (step-2-thres.)	25.64	26.63	192	176	50.56	52.15	28.77	30.21	14.87	14.94
TransEP (step-3-thres.)	24.46	25.36	194	176	47.65	49.45	27.4	28.69	13.56	14.16

Unfortunately, we observe that our model TransEP underperforms on FB15K. TransE model containing only direct relations outperforms in results without type constraints on all metrics. However, type constraints results of TransE containing also reverse relations and 2-steps paths outperform in *hits@k* and Mean Reciprocal Rank, but TransE on the original KG has lower Mean Rank in all cases. Furthermore, TransE step-2 has a better performance in triple classification accuracy.

TransE and TransEP of considering relation paths with at most 3-step achieve worse results. This indicates that it may be unnecessary to consider those relation paths that are too long.

Table 6.3: FB15K Triple Classification Accuracy

Method	Triple Classification Accuracy
TransE	84.01
TransE (step-2)	**85.27**
TransE (step-2-thres.)	84.96
TransE (step-3-thres.)	83.66
TransEP	83.17
TransEP (step-2)	83.82
TransEP (step-2-thres.)	84.26
TransEP (step-3-thres.)	82.74

6.2 Results on SNOMED CT

The following tables 6.4, 6.5 and 6.6 display the results on SNOMED CT for all metrics. We observe that: (1) training on KG, which contain reverse relations and 2-step paths, outperforms training on KG with only direct relations. (2) The *filtered* setting in TransEP results does not provide a significant improvement in contrast with TransE model.

From Table 6.4 we observe that: (1) TransE step-2 has a better performance in *hits@k* and Mean Reciprocal Rank. (2) TransEP step-2 outperforms in Mean Rank. (3) TransEP and TransE step-2 achieve comparable results in *raw* setting.

From Tables 6.5 and 6.6 we observe that: (1) the *raw* setting in TransEP step-2 provides better results for all metrics. (2) The *filtered* setting in TransE significantly outperforms TransEP. (3) TransE step-2 has a better performance for all *filtered* metrics. (4) TransEP step-2 achieves the highest triple classification accuracy, but comparable to the one of TransE step-2, step-3 and TransEP step-3.

As in FB15K, TransE and TransEP on KGs with at most 3-step achieve worse results and we do not consider longer relation paths.

Table 6.4: SNOMED CT no type constraint results

Method	MRR(%)		MR		Hits@10(%)		Hits@3(%)		Hits@1(%)	
	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter
TransE	13.37	17.71	13615	13145	25.21	27.96	15.75	19.56	7.37	12.36
TransE (step-2)	**13.93**	**22.63**	11721	11118	**27.46**	**35.84**	**15.87**	**25.14**	**7.41**	**15.71**
TransE (step-3-thres.)	10.74	21.02	12986	12283	23.05	34.1	11.8	23.17	4.92	14.35
TransEP	10.21	10.5	14007	13901	20.83	21.11	11.76	12.12	4.84	5.09
TransEP (step-2)	13.67	14.21	**10902**	**10796**	26.9	27.33	15.47	16.08	7.23	7.77
TransEP (step-3-thres.)	10.72	11.06	12374	12267	22.62	23.0	11.6	12.0	5.13	5.42

Table 6.5: SNOMED CT type constraint results

Method	MRR(%)		MR		Hits@10(%)		Hits@3(%)		Hits@1(%)	
	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter
TransE	13.68	18.12	6587	6117	25.54	28.42	16.07	19.99	7.66	12.75
TransE (step-2)	14.05	**22.96**	5298	**4695**	27.62	**36.24**	15.99	**25.47**	7.49	**16.01**
TransE (step-3-thres.)	10.84	21.26	5682	4978	23.23	34.51	11.93	23.51	4.98	14.53
TransEP	11.3	11.62	6131	6024	22.14	22.44	13.01	13.4	5.76	6.05
TransEP (step-2)	**15.82**	16.35	**4928**	4823	**28.78**	29.19	**17.65**	18.24	**9.5**	10.02
TransEP (step-3-thres.)	10.81	11.15	5300	5194	22.79	23.17	11.68	12.09	5.17	5.46

Table 6.6: SNOMED CT Triple Classification Accuracy

Method	Triple Classification Accuracy
TransE	86.51
TransE (step-2)	89.67
TransE (step-3-thres.)	89.04
TransEP	84.9
TransEP (step-2)	**89.99**
TransEP (step-3-thres.)	89.17

Chapter 7

Discussion

7.1 Conclusion

In this diploma thesis, we introduced a new Translational Distance Model, TransEP. Our motivation was the prediction of the existence or probability of correctness of triples in a KG, i.e. Link Prediction. We combined TransE and PTransE models by adding multiple-steps paths and reverse relations to the KG, but without impairing the time complexity of TransE. Furthermore, we modified a margin-based loss function by taking into consideration the reliability score of each relation or path in order to consider a triple more if it has larger information content.

We evaluated TransE and TransEP models on two data sets, FB15K and SNOMED CT. We implemented experiments on KGs not only with direct relations, but also with reverse relations and multiple-step relation paths.

With regard to results on FB15K, multiple-step paths do not improve significantly the performance, especially on TransEP, compared to the original data set. Probably, this is due to the fact that FB15K contains thousands of relation types pertaining to world-knowledge without any particular subject. Therefore, the relation paths may not provide any useful information.

On the other hand, concerning SNOMED CT, training on KG with multiple-step paths outperforms the one on KG with only direct relations. Both TransE and our model, TransEP, achieve higher performance on KG with 2-step paths and reverse relations, while TransEP provides better results for some metrics. This significant improvement is probably due to the OWL axioms, on which SNOMED CT is based, such as relations' properties, like the transitivity one, that indicate new important facts.

On both data sets considering relation paths with at most 3-step achieve worse results and we do not consider relation paths that are too long.

7.2 Future Work

There are many avenues to explore future work, but it chiefly could be carried out in two main directions:

- We could use another method to evaluate how important a path is instead of PTransE's reliability score. For example, a possible method could be the use of the PageRank algorithm [4], which gives some approximation of a path's importance or quality and measures the transitive influence or connectivity of nodes in a graph. According to Google:

"PageRank works by counting the number and quality of links to a page to determine a rough estimate of how important the website is. The underlying assumption is that more important websites are likely to receive more links from other websites".

Moreover, Neo4j, which we used to construct the KGs, includes the PageRank algorithm in the Neo4j Graph Algorithms library.

- As TransE and, hence, TransEP have flaws in dealing with 1-to-N, N-to-1 and N-to-N relations, we could rely on another transnational distance model. For instance, TransH overcomes these disadvantages and has the same time complexity with TransE. We could add in the KG the multiple-step paths and the reverse relations, calculate the reliability of each path and modify the loss function of TransH with respect to reliability scores.

Appendices

Appendix A

Mathematics

Definition of a norm We define $\|\mathbf{x}\|_p$ as a "p-norm". Given $\mathbf{x}$, a vector with i components, a p-norm is defined as:

$$\|\mathbf{x}\|_p = \Big(\sum_i \mid x_i \mid^p \Big)^{1/p} \tag{A.1}$$

L1 Norm L1 Norm, also known as Manhattan Distance, is the sum of absolute difference of the components of the vectors and is defined as:

$$\|\mathbf{x}\|_1 = \sum_i \mid x_i \mid \tag{A.2}$$

L2 Norm L2 Norm, also known as Euclidean Distance, is defined as:

$$\|\mathbf{x}\|_2 = \sqrt{\Big(\sum_i \mid x_i \mid^2 \Big)} \tag{A.3}$$

Frobenius Norm The Frobenius norm is matrix a norm of an $m\times$ matrix $\mathbf{A}$ defined as the square root of the sum of the absolute squares of its elements:

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} \mid a_{ij} \mid^2} \tag{A.4}$$

Appendix B

Figures

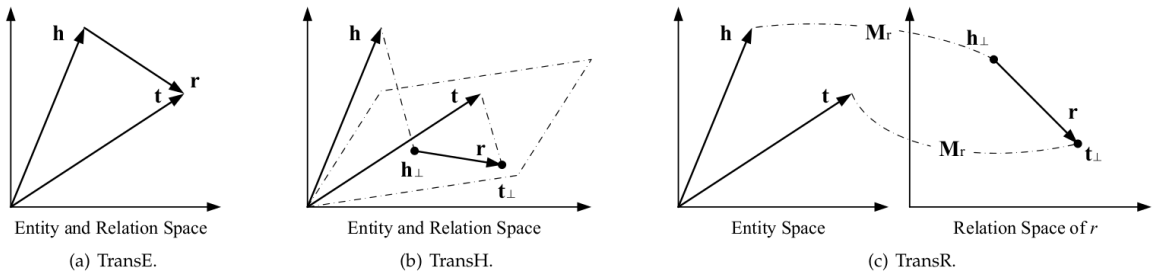


Figure B.1: Simple Illustrations of TransE, TransH, TransR. The figures are adapted from [19, 33, 34]

.

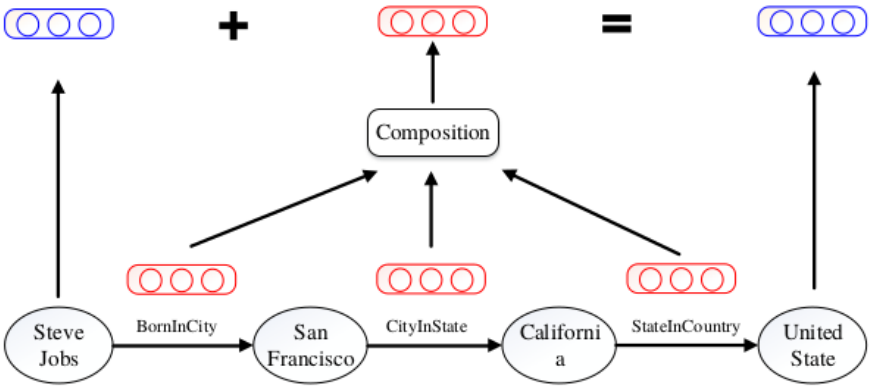


Figure B.2: Path representations are computed by semantic composition of relation embeddings. The figure is adapted from [18]

.

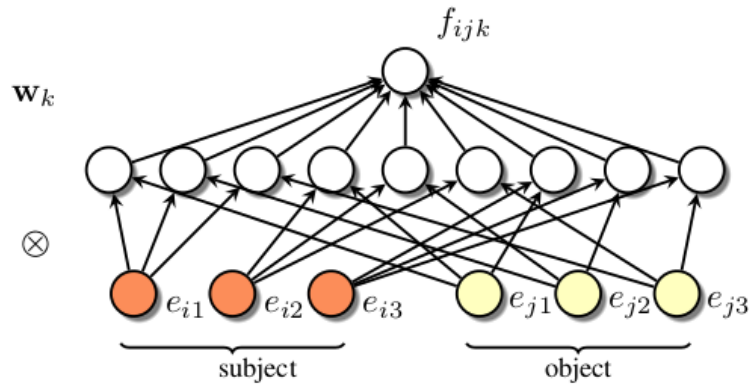


Figure B.3: Visualization of RESCAL as Neural Network. The figure is adapted from [23]

.

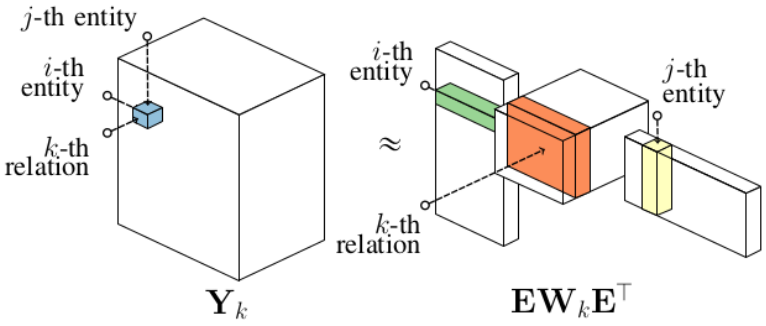


Figure B.4: RESCAL as a tensor factorization of adjacency tensor Y. The figure is adapted from [23]

.

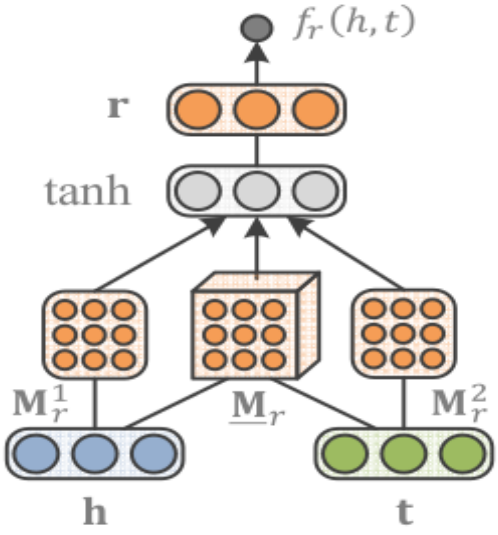


Figure B.5: Neural Network architecture of NTN. The figure is adapted from [29, 33]

.

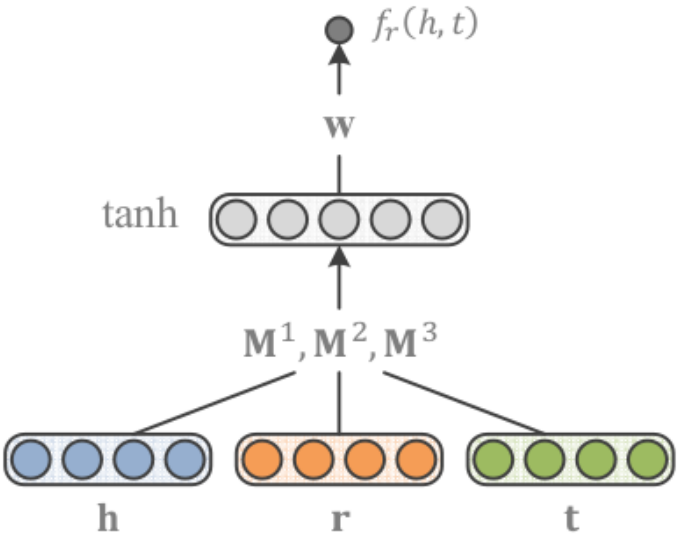


Figure B.6: Neural Network architecture of MLP. The figure is adapted from [33]

.

Bibliography

[1] SNOMED CT. https://en.wikipedia.org/wiki/SNOMED_CT. Accessed: 2019-06-20.

[2] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. pages 2787–2795, 2013.

[3] Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. Learning structured embeddings of knowledge bases. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

[4] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117, 1998.

[5] Tensorflow Community. Embeddings. https://www.tensorflow.org/guide/embedding. Accessed: 2019-05-31.

[6] Rory Davidson. SNOMED CT database. https://github.com/rorydavidson/SNOMED-CT-Database.git.

[7] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 601–610. ACM, 2014.

[8] Lucas Drumond, Steffen Rendle, and Lars Schmidt-Thieme. Predicting rdf triples in incomplete knowledge bases with tensor factorization. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, SAC '12, pages 326–331, New York, NY, USA, 2012. ACM.

[9] Lise Getoor and Lilyana Mihalkova. Learning statistical models from relational data. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pages 1195–1198. ACM, 2011.

[10] Xu Han, Shulin Cao, Xin Lv, Yankai Lin, Zhiyuan Liu, Maosong Sun, and Juanzi Li. OpenKE: An open toolkit for knowledge embedding. In *Proceedings of EMNLP*, pages 139–144, 2018.

[11] Xu Han, Shulin Cao, Lv Xin, Yankai Lin, Zhiyuan Liu, Maosong Sun, and Juanzi Li. OpenKE PyTorch database. https://github.com/thunlp/OpenKE/tree/OpenKE-PyTorch.

[12] Yi Huang, Volker Tresp, Maximilian Nickel, Achim Rettinger, and Hans-Peter Kriegel. A scalable approach for statistical learning in semantic graphs. *Semantic Web*, 5(1):5–22, 2014.

[13] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 687–696, 2015.

[14] Xueyan Jiang, Volker Tresp, Yi Huang, and Maximilian Nickel. Link prediction in multi-relational graphs using additive models. In *Proceedings of the 2012 International Conference on Semantic Technologies Meet Recommender Systems & Big Data - Volume 919*, SeRSy'12, pages 1–12, Aachen, Germany, Germany, 2012. CEUR-WS.org.

[15] Xueyan Jiang, Volker Tresp, Yi Huang, and Maximilian Nickel. Link prediction in multi-relational graphs using additive models. *SeRSy*, 919:1–12, 2012.

[16] Ni Lao and William W. Cohen. Relational retrieval using a combination of path-constrained random walks. *Mach. Learn.*, 81(1):53–67, October 2010.

[17] Ni Lao, Tom Mitchell, and William W. Cohen. Random walk inference and learning in a large scale knowledge base. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 529–539, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[18] Yankai Lin, Zhiyuan Liu, and Maosong Sun. Modeling relation paths for representation learning of knowledge bases. *CoRR*, abs/1506.00379, 2015.

[19] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 2181–2187. AAAI Press, 2015.

[20] Gengchen Mai, Krzysztof Janowicz, and Bo Yan. Support and centrality: Learning weights for knowledge graph embedding models. In *European Knowledge Acquisition Workshop*, pages 212–227. Springer, 2018.

[21] Deborah L McGuinness, Frank Van Harmelen, et al. OWL web ontology language overview. *W3C recommendation*, 10(10):2004, 2004.

[22] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[23] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs, 2015.

[24] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pages 809–816, USA, 2011. Omnipress.

[25] IHTSDO (International Health Terminology Standards Development Organization). SNOMED CT document library: https://confluence.ihtsdotools.org/display/DOC. Accessed: 2019-06-20.

[26] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier, 2014.

[27] Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine learning*, 62(1-2):107–136, 2006.

[28] Corbin L. Rosset. Knowledge base completion with embeddings of entities and relation operations. Master's thesis, Computer Science Department of The Johns Hopkins University, 2017.

[29] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning with neural tensor networks for knowledge base completion. In *Advances in neural information processing systems*, pages 926–934, 2013.

[30] Volker Tresp, Yi Huang, Markus Bundschus, and Achim Rettinger. Materializing and querying learned knowledge. *Proc. of IRMLeS*, 2009, 2009.

[31] Pierre-Yves Vandenbussche. Translating embeddings (TransE). http://pyvandenbussche.info/2017/translating-embeddings-transe/. Accessed: 2019-06-07.

[32] Ellen M. Voorhees. The trec-8 question answering track report. In *In Proceedings of TREC-8*, pages 77–82, 1999.

[33] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Trans. Knowl. Data Eng.*, 29(12):2724–2743, 2017.

[34] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Twenty-Eighth AAAI conference on artificial intelligence*, 2014.

[35] Baichuan Zhang, Sutanay Choudhury, Mohammad Al Hasan, Xia Ning, Khushbu Agarwal, Sumit Purohit, and Paola Gabriela Pesntez Cabrera. Trust from the past: Bayesian personalized ranking based link prediction in knowledge graphs. *CoRR*, abs/1601.03778, 2016.