

# Αυτόματη διάγνωση ασθενειών με χρήση Μεγάλων Γλωσσικών Μοντέλων και Answer Set Programming

Ιωάννα Γέμου\*  
ioannagemou@upnet.gr

Ευάγγελος Λάμπρου\*  
e.lamprou@upnet.gr

31 Μαΐου 2023

## Περιεχόμενα

<b>1</b>	<b>Περίληψη</b>	<b>3</b>
<b>2</b>	<b>Ανασκόπηση Βιβλιογραφίας</b>	<b>3</b>
2.1	Large Language Models . . . . .	3
2.1.1	Inputs and Input Embeddings . . . . .	4
2.1.2	Positional Encodings . . . . .	6
2.1.3	Encoder . . . . .	6
2.1.4	Decoder . . . . .	6
2.2	Δυσκολίες στην εφαρμογή των LLMs . . . . .	6
2.3	Answer Set Programming . . . . .	7
2.4	Χρήση του ASP στην διάγνωση ασθενειών . . . . .	8
<b>3</b>	<b>Μεθοδολογία</b>	<b>9</b>
3.1	Αρχιτεκτονική . . . . .	9
3.2	Κατασκευή Prompt . . . . .	9
3.3	Κατασκευή Συλλογιστικού Προγράμματος . . . . .	10
3.4	Επεξηγηματικότητα Διαγνώσεων . . . . .	13

---

\*Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών, Πανεπιστήμιο Πατρών

4	Αποτελέσματα	15
5	Συμπεράσματα	16
	Αναφορές	17

# 1 Περίληψη

Η ανάπτυξη αυτόματων συστημάτων πρόβλεψης ασθενειών έχει κρίσιμη σημασία για τη βελτίωση της ανθρώπινης υγείας και την υποστήριξη των ιατρών στην ακριβή διάγνωση και αντιμετώπιση των ασθενειών. Η ακριβής διάγνωση είναι κρίσιμη για την έγκαιρη αντιμετώπιση και αποτελεσματική θεραπεία, ενώ μπορεί επίσης να συμβάλει στην αποφυγή περαιτέρω επιπλοκών και επιβαρυντικών ιατρικών διαδικασιών. Η συμβολική τεχνητή νοημοσύνη έχει ξαναεφαρμοστεί στο πεδίο της ιατρικής [1]. Ωστόσο, εφαρμογή πάντα ήταν περιορισμένη λόγω των σύνθετων βάσεων γνώσης που θα έπρεπε να αναπτυχθούν για τη λειτουργία αυτών των συστημάτων. Η προτεινόμενη μέθοδος, η οποία συνδυάζει τα Large Language Models και το Answer Set Programming (ASP), αποτελεί ένα ισχυρό εργαλείο, καθώς παρέχει ένα αξιόπιστο και γρήγορο μέσο δημιουργίας μιας βάσης γνώσης η οποία στη συνέχεια μπορεί να χρησιμοποιηθεί για την πρόβλεψη ασθενειών.

Η καινοτομία της προτεινόμενης μεθόδου έγκειται στον συνδυασμό των δύο μεθοδολογιών. Τα Large Language Models μπορούν να κωδικοποιήσουν την ιατρική βιβλιογραφία σε κώδικα ASP με αξιοσημείωτη ακρίβεια και σύνθετη κατανόηση, ενώ το ASP παρέχει τον συλλογισμό και την επεξηγηματικότητά του. Ο συνδυασμός αυτών των δύο προσεγγίσεων επιτρέπει την ανάπτυξη ενός ισχυρού συστήματος πρόβλεψης ασθενειών που συνδυάζει την **ακρίβεια και την ευφυΐα των γλωσσικών μοντέλων** με την **επεξηγηματικότητα και την αποδοτικότητα του Answer Set Programming**.

Η μέθοδός μας φαίνεται να έχει υποσχόμενα αποτελέσματα πάνω στο σύνολο των δεδομένων ασθενών, στο οποίο την εφαρμόσαμε.

## 2 Ανασκόπηση Βιβλιογραφίας

### 2.1 Large Language Models

Τα μεγάλα γλωσσικά μοντέλα [2] είναι προηγμένα υπολογιστικά μοντέλα που έχουν σχεδιαστεί για να κατανοούν και να δημιουργούν ανθρώπινη γλώσσα. Αυτά τα μοντέλα, που αναπτύχθηκαν χρησιμοποιώντας εξελιγμένες τεχνικές μηχανικής μάθησης, έχουν τη δυνατότητα να επεξεργάζονται και να δημιουργούν κείμενο με τρόπο που μοιάζει πολύ με τα πρότυπα ανθρώπινης γλώσσας.

Μεγάλα γλωσσικά μοντέλα, όπως η αρχιτεκτονική *GPT-4* [3], εκπαιδεύονται σε τεράστιες ποσότητες δεδομένων κειμένου, γεγονός που τους επιτρέπει να αποκτούν γνώση σε διάφορους τομείς και θέματα. Αυτά τα μοντέλα χρησιμοποιούν αλγόριθμους βαθιάς μάθησης (deep learning) [4] και ειδικά δίκτυα μετασχηματιστών (transformer

networks) [5], για την ανάλυση και την κατανόηση των σχέσεων μεταξύ λέξεων, φράσεων και προτάσεων. Καταγράφοντας μοτίβα και δομές στα δεδομένα εκπαίδευσης, αυτά τα μοντέλα μπορούν να δημιουργήσουν συνεκτικές και συναφείς απαντήσεις με βάση δοθέντα prompts.

Τα μοντέλα ονομάζονται “μεγάλα” επειδή έχουν δισεκατομμύρια παραμέτρους που διαμορφώνουν τις αποκρίσεις τους. Η βασική προϋπόθεση ενός γλωσσικού μοντέλου είναι η ικανότητά του να προβλέπει την επόμενη λέξη (που ονομάζεται token) με βάση το κείμενο που έχει παρατηρήσει μέχρι τώρα, δηλαδή σύμφωνα με τα συμφραζόμενα.

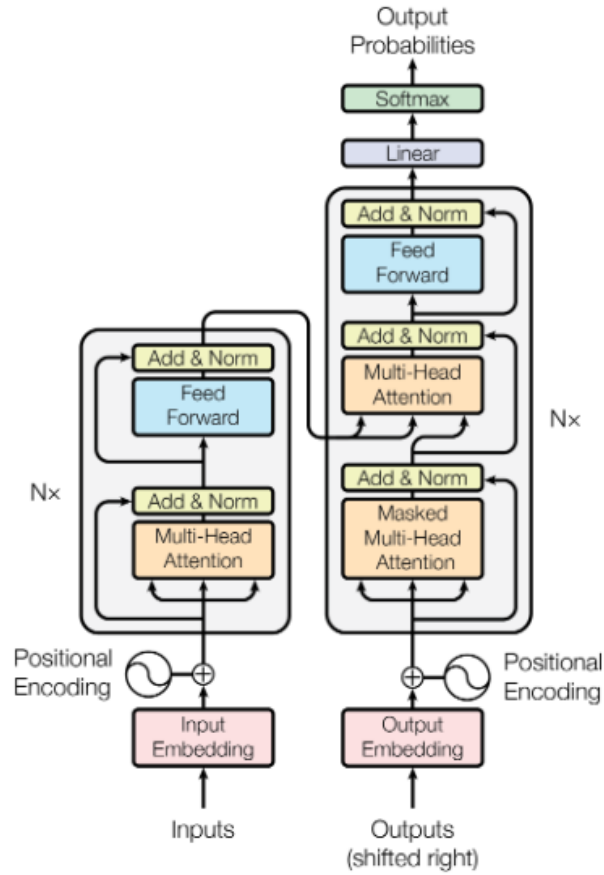
Τα μεγάλα γλωσσικά μοντέλα διαθέτουν την αξιοσημείωτη ικανότητα να μετασχηματίζουν κείμενο από τη μια μορφή στην άλλη, καθιστώντας τα ισχυρά εργαλεία για διάφορες εργασίες επεξεργασίας γλώσσας. Πάνω σε αυτό το χαρακτηριστικό τους, βασίζεται και η ιδέα αυτής της εργασίας, αφού ένα πολύ σημαντικό κομμάτι που πραγματευόμαστε είναι η κατάλληλη κωδικοποίηση της ιατρικής βιβλιογραφίας. Παρακάτω αναφέρουμε κάποια παραδείγματα χρήσης των μεγάλων γλωσσικών μοντέλων [6]:

- Δημιουργία κειμένου που είναι συναφές και συνεκτικό με βάση τα συμφραζόμενα
- Μετάφραση Γλώσσας
- Σύνοψη / Περίληψη Κειμένου
- Απάντηση σε ερωτήσεις
- Παραγωγή κειμένου με βάση συγκεκριμένων κανόνων που τους παρέχονται

Ο θεμέλιος λίθος των Μεγάλων Γλωσσικών μοντέλων είναι οι **transformers**. Οι transformers [7] επί του παρόντος θεωρούνται μία από τις πιο προηγμένες και ισχυρές κατηγορίες μοντέλων. Συνέβαλαν καθοριστικά στην επίτευξη σημαντικών ανακαλύψεων στη μηχανική μάθηση, με αποτέλεσμα ορισμένοι να τους αναφέρουν ως “transformers AI”. Αυτά τα μοντέλα οδηγούν σε μια θεμελιώδη αλλαγή στον τομέα της τεχνητής νοημοσύνης [8]. Το σχήμα 1 απεικονίζει τη δομή του transformer, ο οποίος περιλαμβάνει δύο διακριτά μέρη, την κωδικοποίηση και την αποκωδικοποίηση.

### 2.1.1 Inputs and Input Embeddings

Το θεμελιώδες συστατικό του transformer, ο μηχανισμός “προσοχής” είναι εμπνευσμένος από την έννοια της συνειδητής προσοχής στους ζωντανούς οργανισμούς [7]. Αυτός ο μηχανισμός επιτρέπει στο δίκτυο να αποδίδει μεγαλύτερη σημασία σε κρίσιμα μέρη των δεδομένων και λιγότερη σημασία σε αυτά που είναι λιγότερο σχετικά. Αν και η απλούστερη μορφή αυτής της συνάρτησης είναι ένα σταθμισμένο άθροισμα



Σχήμα 1: Αρχιτεκτονική του μοντέλου transformer [7].

των τιμών εισόδου, ο μηχανισμός προσοχής στον μετασχηματιστή είναι πιο περίπλοκος και περιλαμβάνει πολλαπλούς μετασχηματισμούς για τον υπολογισμό των βαρών προσοχής μεταξύ διαφορετικών τμημάτων της εισόδου.

Τα τμήματα εισόδου από τον χρήστη είναι είσοδοι για τα μοντέλα μηχανικής μάθησης. Ωστόσο, τα μοντέλα κατανοούν μόνο αριθμούς, όχι κείμενο, επομένως αυτές οι είσοδοι πρέπει να μετατραπούν σε μια αριθμητική μορφή που ονομάζεται **input embeddings**. Τα input embeddings αντιπροσωπεύουν λέξεις ως αριθμούς, τους οποίους μπορούν στη συνέχεια να επεξεργαστούν τα μοντέλα μηχανικής μάθησης. Τα embeddings είναι σαν ένα λεξικό, το οποίο βοηθά το μοντέλο να κατανοήσει τη σημασία των λέξεων, μέσω της τοποθέτησής τους σε έναν μαθηματικό χώρο όπου παρόμοιες λέξεις βρίσκονται η μία κοντά στην άλλη. Κατά τη διάρκεια της εκπαίδευ-

σης, το μοντέλο μαθαίνει πώς να δημιουργεί τα embeddings, έτσι ώστε παρόμοια διανύσματα να αντιπροσωπεύουν λέξεις με παρόμοια σημασία.

### 2.1.2 Positional Encodings

Στην επεξεργασία φυσικής γλώσσας, η σειρά των λέξεων σε μια πρόταση είναι κρίσιμη για τον προσδιορισμό του νοήματος της πρότασης. Ωστόσο, τα παραδοσιακά μοντέλα μηχανικής μάθησης, όπως τα νευρωνικά δίκτυα, δεν κατανοούν εγγενώς τη σειρά των εισόδων. Για την αντιμετώπιση αυτής της πρόκλησης, η κωδικοποίηση θέσης μπορεί να χρησιμοποιηθεί για την κωδικοποίηση της θέσης κάθε λέξης σε μία πρόταση. Η κωδικοποίηση αυτή τροφοδοτείται στο μοντέλο Transformer, μαζί με τα input embeddings. Με την ενσωμάτωση της κωδικοποίησης θέσης στην αρχιτεκτονική του Transformer, το μοντέλο μπορεί να κατανοήσει πιο αποτελεσματικά τη σειρά των λέξεων σε μια πρόταση και να δημιουργήσει γραμματικά σωστά και σημασιολογικά νόημα. Το “γραμματικά σωστό” αποτέλεσμα μπορεί να σημαίνει και την παραγωγή κώδικα ο οποίος είναι σύμφωνος με τους κανόνες μιας γλώσσας προγραμματισμού.

### 2.1.3 Encoder

Ο κωδικοποιητής (encoder) είναι μέρος του νευρωνικού δικτύου που επεξεργάζεται το κείμενο εισόδου και δημιουργεί μια σειρά από κρυφές καταστάσεις που καταγράφουν το νόημα του κειμένου. Ο κωδικοποιητής διαμορφώνει πρώτα το κείμενο εισόδου σε μια ακολουθία tokens, π.χ. μεμονωμένες λέξεις.

### 2.1.4 Decoder

Ο αποκωδικοποιητής (decoder) είναι μέρος του μοντέλου που δημιουργεί την ακολουθία εξόδου με βάση την κωδικοποιημένη ακολουθία εισόδου. Κατά τη διάρκεια της εκπαίδευσης, ο αποκωδικοποιητής μαθαίνει πώς να “μαντεύει” την επόμενη λέξη κοιτάζοντας τις λέξεις πριν από αυτήν. Για παράδειγμα, ο αποκωδικοποιητής σε *GPT* δημιουργεί κείμενο φυσικής γλώσσας με βάση την ακολουθία εισόδου και το περιβάλλον που μαθαίνει ο κωδικοποιητής. Στους transformers χρησιμοποιούνται πολλαπλά στρώματα αποκωδικοποιητών.

## 2.2 Δυσκολίες στην εφαρμογή των LLMs

Τα μεγάλα γλωσσικά μοντέλα δεν είναι αλάνθαστα [9], [10]. Μπορούν περιστασιακά να παράγουν λανθασμένες ή μεροληπτικές απαντήσεις, αντανακλώνοντας τις προκαταλήψεις που υπάρχουν στα δεδομένα εκπαίδευσης. Καταβάλλονται προσπάθειες

για τον μετριασμό αυτών των ζητημάτων μέσω τεχνικών προεπεξεργασίας δεδομένων και fine-tuning.

Συνολικά, τα μεγάλα γλωσσικά μοντέλα έχουν επιδείξει τεράστιες δυνατότητες σε ένα ευρύ φάσμα εφαρμογών, συμπεριλαμβανομένης της κατανόησης φυσικής γλώσσας, της δημιουργίας περιεχομένου και της αλληλεπίδρασης ανθρώπου-υπολογιστή. Η συνεχής έρευνα και ανάπτυξη σε αυτόν τον τομέα υπόσχεται περαιτέρω ενίσχυση αυτών των μοντέλων και των εφαρμογών τους σε διάφορους τομείς, συμβάλλοντας στην πρόοδο των τεχνολογιών γλωσσικής επεξεργασίας και επικοινωνίας.

## 2.3 Answer Set Programming

Το Answer Set Programming (ASP) [11] είναι μια δηλωτική μέθοδος επίλυσης προβλημάτων που βασίζεται στον Λογικό Προγραμματισμό και τη Μη Μονοτονική Συλλογιστική. Στην εργασία [12] επισημοποιήθηκε αρχικά η σημασιολογία των σταθερών μοντέλων και η βασική γλώσσα του ASP.

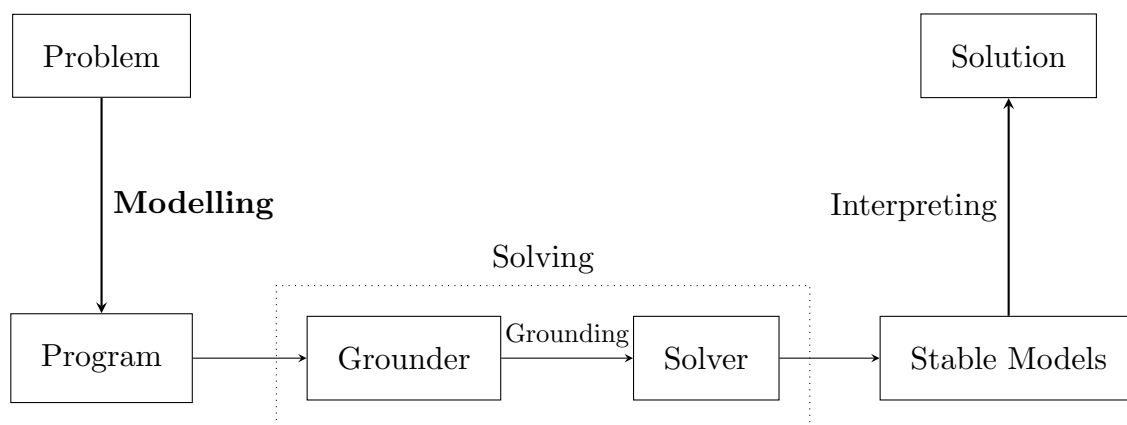
Ο προγραμματισμός με χρήση αυτής της προσέγγισης γίνεται σε μια οικογένεια γλωσσών που ονομάζεται ορισμένες φορές *AnsProlog* [13].

Η ιδέα πίσω από το ASP είναι να μοντελοποιηθεί ένα πρόβλημα ως ένα σύνολο κανόνων και δεδομένων, και στη συνέχεια να χρησιμοποιηθεί ένας επίλυτης για να βρεθεί μια λύση για το πρόβλημα. Οι λύσεις παριστάνονται από σταθερά μοντέλα ή σύνολα απαντήσεων. Οι κανόνες, τα δεδομένα και οι περιορισμοί που περιγράφουν το πρόβλημα αποτελούν τα στοιχεία του προγράμματος. Στη συνέχεια, το πρόγραμμα δίνεται σε έναν επίλυτη που θα βρει μία ή περισσότερες λύσεις.

Στον παραδοσιακό προγραμματισμό, η μετάβαση από ένα πρόβλημα σε μια λύση προϋποθέτει την απόλυτη κατανόηση του δεδομένου προβλήματος από τον προγραμματιστή, και έπειτα την παραγωγή ενός προγράμματος, στο οποίο όταν δοθεί ένα instance του προβλήματος θα παράγει το σωστό αποτέλεσμα, τα οποία στη συνέχεια θα ερμηνευτεί ως λύση.

Στο ASP, η μετάβαση από τη δήλωση προβλήματος σε ένα σύνολο λύσεων περιλαμβάνει τα ακόλουθα βήματα [14] (εικόνα 2):

- **Modeling:** Το πρόβλημα μοντελοποιείται σε γλώσσα ASP.
- **Grounding:** Ένας grounder (π.χ. gringo) μετατρέπει το πρόγραμμα σε ένα σύνολο βασικών κανόνων και δεδομένων.
- **Solving:** Ένας επίλυτης (π.χ. clasp) βρίσκει μια λύση στο πρόβλημα υπολογίζοντας το σύνολο των σταθερών μοντέλων (σύνολα απαντήσεων).



Σχήμα 2: Η επίλυση ενός προβλήματος με χρήση ASP.

Έχουν αναπτυχθεί αρκετοί λύτες για ASP όπως το DLV [15]. Σε αυτή την εργασία, θα εφαρμόσουμε το σύστημα ASP clingo [16] το οποίο συνδυάζει τον grounder gringo με τον clasp solver [17] σε μια ενιαία εφαρμογή, παρέχοντας επίσης ένα ισχυρό Python API για την ενσωμάτωση του solver σε άλλες εφαρμογές.

## 2.4 Χρήση του ASP στην διάγνωση ασθενειών

Το Answer Set Programming (ASP) προσφέρει μια ισχυρή μέθοδο για την αναπαράσταση της ιατρικής γνώσης με δηλωτικό και λογικό τρόπο. Ακολουθούν ορισμένες μέθοδοι ASP που χρησιμοποιούνται συνήθως στην αναπαράσταση ιατρικής γνώσης [18]:

- **Αναπαράσταση βάσει κανόνων:** Το ASP επιτρέπει την αναπαράσταση της ιατρικής γνώσης ως κανόνων με τη μορφή λογικών δηλώσεων. Αυτοί οι κανόνες ορίζουν σχέσεις μεταξύ ιατρικών οντοτήτων όπως συμπτώματα, ασθένειες, θεραπείες και δεδομένα ασθενών. Για παράδειγμα, ένας κανόνας θα μπορούσε να αναφέρει ότι εάν ένας ασθενής εμφανίζει συγκεκριμένα συμπτώματα, υποδηλώνει την παρουσία ή την απουσία ορισμένων ασθενειών.
- **Κατασκευή Γνωσιακής Βάσης:** Το ASP παρέχει ένα πλαίσιο για την κατασκευή μιας βάσης γνώσεων που συλλαμβάνει την ιατρική γνώση. Η βάση γνώσεων αποτελείται από γεγονότα, κανόνες και περιορισμούς που καθορίζουν τις πληροφορίες και τους περιορισμούς που σχετίζονται με την ιατρική διάγνωση και θεραπεία. Μπορεί να περιλαμβάνει πληροφορίες σχετικά με τις σχέ-



σεις ασθένειας-συμπτωμάτων, διαγνωστικά κριτήρια, κατευθυντήριες γραμμές θεραπείας και δεδομένα για τον ασθενή.

Στην εργασία [19] εισήχθησαν νέες μέθοδοι για αποτελεσματικό χειρισμό σύνθετων ερωτημάτων στο πλαίσιο βιοϊατρικών οντολογιών και βάσεων δεδομένων, με έμφαση στην εξαγωγή σχετικών πληροφοριών από αυτούς τους πόρους γνώσης. Αυτές οι μέθοδοι εφαρμόστηκαν για την αντιμετώπιση απαιτητικών ερωτημάτων στον τομέα της ανακάλυψης φαρμάκων, χρησιμοποιώντας βιολογικές βάσεις δεδομένων σε συνδυασμό με συνοπτικές εξηγήσεις για πολύπλοκα ερωτήματα που σχετίζονται με διαδικασίες ανακάλυψης φαρμάκων.

## 3 Μεθοδολογία

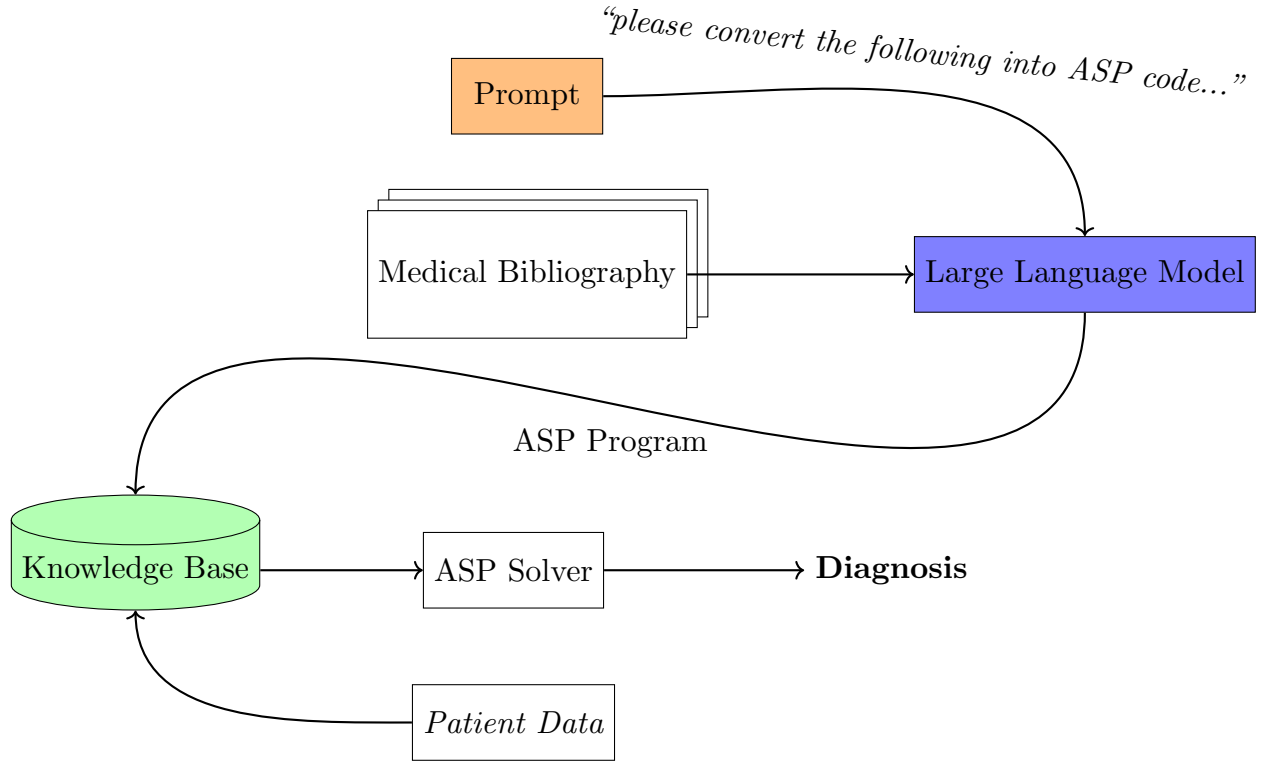
### 3.1 Αρχιτεκτονική

Σκοπός μας είναι η εξαγωγή μίας βάσης γνώμης μέσα από αδόμητο κείμενο [20]. Η αρχιτεκτονική μας (σχήμα 3) περιλαμβάνει τα εξής στάδια

1. Συλλογή ιατρικής βιβλιογραφίας σχετική με τις εξεταζόμενες ασθένειες/παθήσεις.
2. Κατασκευή κατάλληλου prompt για την μετατροπή ιατρικής βιβλιογραφίας σε κώδικα ASP.
3. Προσθήκη του παραγόμενου ASP κώδικα από το μεγάλο μοντέλο γλώσσας στη βάση γνώσης.
4. Προσθήκη σχετικών δεδομένων του ασθενή (συμπτώματα, αποτελέσματα ιατρικών εξετάσεων) στη βάση γνώσης.
5. Παραγωγή της τελικής διάγνωσης (ή σύνολο πιθανών διαγνώσεων) ως answer set(s), αφού ο solver συλλογιστεί πάνω στη τελική βάση γνώσης.

### 3.2 Κατασκευή Prompt

Τα μεγάλα γλωσσικά μοντέλα, παρά των υψηλών δυνατοτήτων τους στην εκτέλεση οδηγιών, ακόμα έχουν χαμηλή ελεγκσιμότητα [21]. Έτσι, έχει προκύψει μία σχετικά νέα μεθοδολογία “προγραμματισμού” γνωστή ως prompt engineering [22, 23]. Προκειμένου να μετατρέψουμε την ιατρική βιβλιογραφία σε κώδικα κατάλληλο για την εκτέλεσή του σε επιλυτή ASP έπρεπε να κατασκευάσουμε ποιοτικά prompts ώστε ο παραγόμενος κώδικας να βρίσκεται στην επιθυμητή μορφή (σχήμα 4, σχήμα 5).



Σχήμα 3: Η διαδικασία της διάγνωσης με τη χρήση της προτεινόμενης μεθόδου. Ιατρική βιβλιογραφία, με χρήση ενός μεγάλου γλωσσικού μοντέλου, μετατρέπεται σε ένα πρόγραμμα ASP. Στη συνέχεια, το πρόγραμμα αυτό εκτελείται από έναν ASP solver, ο οποίος παράγει την τελική διάγνωση με βάση τα δεδομένα για τον εκάστοτε ασθενή.

### 3.3 Κατασκευή Συλλογιστικού Προγράμματος

Η βάση γνώσης αρχικά αποτελείται από ένα σύνολο συμπτωμάτων και ιατρικών ενδείξεων της μορφής:

$$\text{symptom}(s). s \in \{\text{cough, chest pain, rash, ...}\} \quad (1)$$

$$\text{indicator}(i). i \in \{\text{low MCV, high TIBC, ...}\} \quad (2)$$

Αποτυπώνουμε έναν ασθενή ο οποίος έχει κάποιο σύμπτωμα ή βγήκε θετικός σε κάποια εξέταση ως:



**Prompt:** *Medical Text...*

The above paragraph refers some common symptoms of pneumonia. Give me a Clingo script so that based on symptoms I can diagnose pneumonia.

```

LLM Output

% Symptoms of Pneumonia

symptom(cough).
symptom(cough_with_mucus).
symptom(chest_pain).
symptom(headache).
...

% Diagnosis of pneumonia
diagnosis(pneumonia) :- symptom(cough),
                        symptom(chest_pain),
                        symptom(headache),
                        symptom(arthralgia),
                        ...

```

Σχήμα 4: Ένα απλό prompt μέσω του οποίου δεν πέρνουμε ικανοποιητικά αποτελέσματα.

$$has(x). x \in S \cup I \quad (3)$$

Οι λογικές προτάσεις με βάση τις οποίες το σύστημά μας θα βγάζει συμπεράσματα είναι της μορφής:

$$diagnosis(d_1) \leftarrow has(x_1) \wedge has(x_2) \wedge \dots \quad (4)$$

$$diagnosis(d_1) \leftarrow has(x_1) \wedge has(x_3) \wedge \dots \quad (5)$$

$$diagnosis(d_2) \leftarrow has(x_3) \wedge has(x_4) \wedge \dots \quad (6)$$



**Prompt:** *Medical Text...*

The above paragraph refers some common symptoms of pneumonia. Give me a Clingo script so that based on symptoms I can diagnose pneumonia.

In the clingo script that you produced before, make some changes in diagnosis rule like: `diagnosis(pneumonia) :- has(symptom(x)), has(symptom(y))`  
...

Include some alternative diagnosis for pneumonia. Create rules based on one symptom being linked to another.

LLM Output

```
% Symptoms of Pneumonia
symptom(cough).
symptom(cough_with_mucus).
...

% Rules linking symptoms

% Cough with mucus can be associated with wheezing and
% shortness of breath
linked_symptom(cough_with_mucus, wheezing).
linked_symptom(cough_with_mucus, shortness_of_breath).
...

% Rule to infer linked symptoms
has(symptom(Symptom2)) :- has(symptom(Symptom1)),
    linked_symptom(Symptom1, Symptom2).

% Diagnosis of Pneumonia
diagnosis(pneumonia) :- has(symptom(cough)),
    has(symptom(cough_with_mucus)),
...

% Alternative Diagnoses for Pneumonia
diagnosis(pneumonia) :- has(symptom(cough_with_mucus)),
    has(symptom(chest_pain)),
...
```

Σχήμα 5: Δίνοντας αναλυτικότερες πληροφορίες μπορούμε να καθοδηγήσουμε το μεγάλο γλωσσικό μοντέλο στο να μετατρέπει την εισαγόμενη ιατρική βιβλιογραφία στην μορφή που θέλουμε.

Ακόμα, λαμβάνουμε υπόψιν τη συσχέτιση μεταξύ ορισμένων συμπτωμάτων. Παραδείγματος χάρη, σε ορισμένους ασθενείς έχει σημειωθεί σύνδεση μεταξύ δερματικών ασθενειών και πονοκεφάλων [24]. Άλλες φορές, συμπτώματα μπορεί να οδηγήσουν στην εξέλιξη άλλων συμπτωμάτων. Τέτοιου είδους σχέσεις κωδικοποιούνται στη βάση γνώσης μας μέσω κανόνα της μορφής.

$$has(symptom(S2)) \leftarrow has(symptom(S1)) \wedge linked\_symptom(S1, S2). \quad (7)$$

Τα αντίστοιχα δεδομένα θα είναι ως εξής

$$linked\_symptom(grunting, chest\_retractions). \quad (8)$$

$$linked\_symptom(nasal\_flaring, chest\_retractions). \quad (9)$$

Ωστόσο, είναι αναμενόμενο πως κάποιος ασθενής δεν θα κατέχει όλα τα καταγεγραμμένα συμπτώματα. Θέλουμε να δώσουμε έναν ελέγξιμο τρόπο το σύστημά μας να δίνει εναλλακτικές προβλέψεις στην περίπτωση όπου το ιατρικό προσωπικό δεν έχει όλες τις πληροφορίες για την κατάσταση του ασθενή. Έτσι, προσθέτουμε τον παρακάτω κανόνα επιλογής (choice rule), με τον οποίο προστίθεται ένα σύνολο πιθανών συμπτωμάτων για τον ασθενή στη βάση γνώσης.

$$\{add(symptom(S)) : symptom(S)\}. \quad (10)$$

Ακόμα, απαιτούμε από τον επιλυτή να βρει τουλάχιστον μία διάγνωση.

$$\perp \leftarrow \sim diagnosis(\_) \quad (11)$$

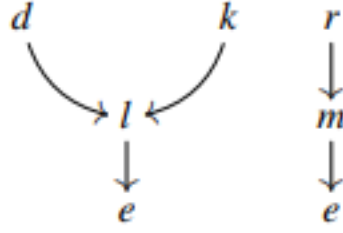
Τέλος, ελαχιστοποιούμε τον αριθμό “ψεύτικων” διαγνώσεων, έτσι ώστε ο επιλυτής να καταλήξει σε μία διάγνωση με όσο το δυνατόν λιγότερες υποθέσεις.

$$\#minimize\{1, S : add(S)\} \quad (12)$$

Με αυτό τον τρόπο, το σύστημά μας εγγυάται πως θα παράξει μία τουλάχιστον πρόβλεψη, η οποία θα βασίζεται όσο το δυνατόν περισσότερο στις έγγυρες πληροφορίες που υπάρχουν για τον ασθενή.

### 3.4 Επεξηγηματικότητα Διαγνώσεων

Η συμβολική τεχνητή νοημοσύνη έχει ένα μείζον πλεονέκτημα έναντι των υποσυμβολικών προσεγγίσεων λόγω της εξηγηματικότητάς της. Οι σχέσεις μεταξύ των



Σχήμα 6: Αιτιολογικός γράφος λογικού προγράμματος (εξ. 13).

γεγονότων και των συμπερασμάτων που καταλήγει ένα σύστημα, το οποίο εκτελεί ένα συμβολικό μηχανισμό είναι σαφώς καθορισμένες. Ωστόσο, οι επίλυτές ASP όπως το *Clingo* δεν παρέχουν<sup>1</sup> εύληπτα αποτυπώματα των συλλογιστικών διαδικασιών τους. Έχει υπάρξει έρευνα στην σύνδεση λογικών προτάσεων με τις αιτιολογήσεις τους [25]. Θεωρήστε το παρακάτω λογικό πρόγραμμα, η σύνδεση των λογικών προτάσεων με τις αιτίες τους παρουσιάζονται σε μορφή γράφου στο σχήμα 6. Αυτή είναι η δομή που μας επιτρέπει να εξηγήσουμε τα συμπεράσματα ενός επιλυτή συμβολικής λογικής.

$$\begin{aligned}
 l : \text{punish} &\leftarrow \text{drive}, \text{drunk} \\
 m : \text{punish} &\leftarrow \text{resist} \\
 e : \text{prison} &\leftarrow \text{punish} \\
 d : \text{drive} \\
 k : \text{drunk} \\
 r : \text{resist}
 \end{aligned} \tag{13}$$

Το εργαλείο *xclingo*<sup>2</sup> [26] επιτρέπει την παρουσίαση αιτιολογήσεων σχετικά με τα αποτελέσματα που δίνονται από τον επιλυτή *Clingo*. Αυτό επιτυγχάνεται προσθέτοντας ένα *trace* που αντιστοιχεί στους κανόνες του λογικού προγράμματός μας. Η παροχή μιας σαφούς εξηγήσεως για το πώς το σύστημα κατέληξε σε συγκεκριμένα συμπεράσματα, μπορεί να βοηθήσει στον εντοπισμό πιθανών σφαλμάτων κατά την δημιουργία της βάσης γνώσης και να παρέχει στον χρήστη του συστήματος μια αιτιολόγηση για το τελικό αποτέλεσμα. Αυτό είναι κάτι πολύ σημαντικό στην περίπτωση που η εφαρμογή χρησιμοποιείται σε ιατρικά πλαίσια.

<sup>1</sup>Τη στιγμή της συγγραφής της εργασίας αυτής.

<sup>2</sup><https://github.com/bramucas/xclingo2>

## 4 Αποτελέσματα

Για την αξιολόγηση του συστήματός μας χρησιμοποιήσαμε δημοσίως διαθέσιμα δεδομένα ασθενών<sup>3</sup>. Στο σύστημά μπήκε σαν είσοδος το σύνολο των συμπτωμάτων του κάθε ασθενή. Σαν βάση γνώσης έχουμε τον συνδυασμό όλων των κανόνων που έχουν προκύψει από την κωδικοποίηση την ιατρικής βιβλιογραφίας. Μία επιτυχημένη πρόβλεψη ορίζεται όταν η πραγματική ασθένεια του ασθενή συμπίπτει με την τελική πρόβλεψη του επιλυτή ASP.

Ασθένειες	Μέγεθος Βάσης Γνώσης	Ακρίβεια
Chicken Pox	66	95%
Pneumonia	75	100%
Common Cold	44	100%

Πίνακας 1: Ακρίβεια της μεθόδου για μερικές ενδεικτικές ασθένειες.

Η ακρίβεια του μοντέλου μας μπορεί να φτάσει πολύ υψηλά ποσοστά ακρίβειας αλλά απαιτεί την εισαγωγή ενός μεγάλου πλήθους κανόνων στη βάση γνώσης. Το μέγεθος της βάσης γνώσης το ορίζουμε ως το πλήθος των λογικών όρων μέσα στο ASP πρόγραμμα που αντιστοιχεί σε κάθε μία από τις ασθένειες.

Στο πρόγραμμα που κατασκευάσαμε προσθέσαμε και τη δυνατότητα επεξήγησης της διάγνωσης, κατά την οποία αξιοποιούμε το εργαλείο `xclingo` για τη δημιουργία ενός γραφήματος το οποίο αποτυπώνει τους συνειρμούς που ακολούθησε το σύστημά μας για τις προβλέψεις του.

```
*
|__diagnosis(chickenpox)
|  |__has(symptom(itching));has(symptom(itching))
|  |__has(symptom(fatigue));has(symptom(fatigue));has(symptom(fatigue
|  |__has(symptom(lethargy))
|  |  |__has(symptom(fatigue));has(symptom(fatigue));has(symptom(
|  |  |__linked_symptom(fatigue,lethargy)
|  |  |__has(symptom(high_fever))
|  |  |__has(symptom(mild_fever));has(symptom(mild_fever))
|  |  |  |__has(symptom(loss_of_appetite));has(symptom(
|  |  |  |__linked_symptom(loss_of_appetite,mild_fever)
```

<sup>3</sup><https://www.kaggle.com/datasets/itachi9604/disease-symptom-description-dataset?select=dataset.csv>

```

| | |__linked_symptom(mild_fever,high_fever)
| | |__has(symptom(loss_of_appetite));has(symptom(loss_of_appetite))
| | |__has(symptom(mild_fever));has(symptom(mild_fever))
| | |__has(symptom(loss_of_appetite));has(symptom(loss_of_appetite))
| | )
| | |__linked_symptom(loss_of_appetite,mild_fever)
| | |__has(symptom(swelled_lymph_nodes))
| |
...

```

Καταχώριση 1: Μέρος της επεξήγησης για τη διάγνωση της ασθένειας chickenpox. Μέσα στο διάγραμμα δέντρο φαίνονται οι συσχετίσεις μεταξύ των διαφορετικών συμπτωμάτων και το πώς το σύστημά μας καταλήγει στα συμπεράσματά του.

## 5 Συμπεράσματα

Στην εργασία αυτή έγινε η παρουσίαση μίας μεθοδολογίας για την αυτόματη πρόβλεψη ασθενειών η οποία περιλαμβάνει την κωδικοποίηση ιατρικής βιβλιογραφίας και τη μετατροπή της σε κώδικα ASP μέσω ενός μεγάλου γλωσσικού μοντέλου. Έγινε παρουσίαση του τρόπου λειτουργίας των μεγάλων γλωσσικών μοντέλων και του ASP, ενώ έγινε συγκεκριμένη αναφορά στο ρόλο που παίζει η κάθε τεχνολογία στο τελικό σύστημα. Ακόμα, δείξαμε την αρχιτεκτονική μας και το πώς αυτές οι τεχνολογίες θα συνδυαστούν ενώ δώσαμε συγκεκριμένες υλοποιητικές λεπτομέρειες τις οποίες αναπτύξαμε τόσο για την δημιουργία ποιοτικών prompts για το μεγάλο γλωσσικό μοντέλο, όσο και στην δομή των κανόνων στην τελική ASP βάση γνώσης. Τελικά, εκτελέσαμε προβλέψεις πάνω σε ορισμένες ασθένειες για τις οποίες δημιουργήσαμε την βάση γνώσης με τη χρήση της μεθοδολογίας μας και παρουσιάσαμε τα αποτελέσματα. Η μέθοδός μας φαίνεται να έχει υποσχόμενα αποτελέσματα και προσφέρει αξιόπιστες προβλέψεις υψηλής ακρίβειας μαζί με τις επεξηγήσεις τους.

Μελλοντική δουλειά είναι η επέκταση της μεθόδου για περισσότερες ασθένειες. Ταυτόχρονα θα ήταν χρήσιμη η δημιουργία μίας γραφικής διεπαφής μέσω της οποίας κάποιος να μπορεί να εισάγει στοιχεία κάποιου ασθενή και να παίρνει πιθανές διαγνώσεις.



## Αναφορές

- [1] Mario Alviano, Riccardo Bertolucci, Valeria Cardellini, Carmine Dodaro, Giuseppe Galatà, Muhammad Kashif Khan, Nicola Leone, Marco Maratea, Francesco Ricca, and Marco Schouten. Answer set programming in healthcare: Extended overview. IPS-RCRA@ AI\* IA, 2020.
- [2] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models, 2023.
- [3] OpenAI. Gpt-4 technical report, 2023.
- [4] Iqbal H. Sarker. Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions. SN Computer Science, 2(6), August 2021.
- [5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2020.
- [6] Bhargavi Paranjape, Scott Lundberg, Sameer Singh, Hannaneh Hajishirzi, Luke Zettlemoyer, and Marco Tulio Ribeiro. Art: Automatic multi-step reasoning and tool-use for large language models, 2023.
- [7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc., 2017.
- [8] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby

Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models, 2021.

- [9] Harsh Raj, Domenic Rosati, and Subhabrata Majumdar. Measuring reliability of large language models through semantic consistency, 2023.
- [10] Laura Eline Ruis, Akbir Khan, Stella Biderman, Sara Hooker, Tim Rocktäschel, and Edward Grefenstette. Large language models are not zero-shot communicators, 2023.
- [11] Thomas Eiter, Giovambattista Ianni, and Thomas Krennwallner. Answer set programming: A primer. volume 5689, pages 40–110, 01 2009.
- [12] Michael Gelfond and Vladimir Lifschitz. The stable model semantics for logic programming. Logic Programming, 2, 12 2000.
- [13] Michael Gelfond. Representing knowledge in a-prolog. In Computational Logic: Logic Programming and Beyond, pages 413–451. Springer Berlin Heidelberg, 2002.
- [14] Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub. Answer Set Solving in Practice. Springer International Publishing, 2013.
- [15] Boming Xia, Xiaozhen Ye, and Adnan Abuassba. Recent research on ai in games. pages 505–510, 06 2020.

- [16] Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub. Clingo = asp + control: Preliminary report, 2014.
- [17] Steffen Hölldobler and Lukas Schweizer. Answer set programming and clasp: A tutorial. volume 1145, 04 2014.
- [18] Retno Aulia Vinarti. Knowledge representation for infectious disease risk prediction system: A literature review. Procedia Computer Science, 161:821–825, 2019.
- [19] Esra Erdem, Yelda Erdem, Halit Erdogan, and Umut Oztok. Finding answers and generating explanations for complex biomedical queries. Proceedings of the AAAI Conference on Artificial Intelligence, 25(1):785–790, August 2011.
- [20] Christopher Ré, Amir Abbas Sadeghian, Zifei Shan, Jaeho Shin, Feiran Wang, Sen Wu, and Ce Zhang. Feature engineering for knowledge base construction, 2014.
- [21] Samuel R. Bowman. Eight Things to Know about Large Language Models, 2023.
- [22] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing, 2021.
- [23] Prompt Engineering Guide | Prompt Engineering Guide — [promptingguide.ai](https://www.promptingguide.ai). <https://www.promptingguide.ai/>. [Accessed 27-May-2023].
- [24] Migraine.com Editorial Team. Migraine hives symptoms. <https://migraine.com/migraine-symptoms/hives>. Accessed on May 30, 2023.
- [25] Pedro Cabalar, Jorge Fandinno, and Michael Fink. Causal graph justifications of logic programs, 2014.
- [26] Pedro Cabalar, Jorge Fandinno, and Brais Muñoz. A System for Explainable Answer Set Programming. Electronic Proceedings in Theoretical Computer Science, 325:124–136, sep 2020.