مجلة جامعة طرطوس للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (6) العدد(10) 2022
Tartous University Journal for Research and Scientific Studies - engineering Sciences Series Vol. (6) No. (10) 2022

تحسين خوارزمية FP-growth لتحليل معاني جمل اللغة الإنكليزية بشكل أفضل

أ.د.م .يعرب ديوب [*]
م .ميريام عبيد [**]

(تاريخ الإيداع 2022/8/16 . قُبِل للنشر في 2022/11/6)

□ ملخّص□

يعد تدقيق معاني اللغات الطبيعية من الأهداف الأساسية لعلماء اللغة والمهتمين بعلم اللغات الحاسوبية Computational Linguistics لأنه أصبح من الضروري تدقيق النصوص المكتوبة على الحاسب في مجالات مختلفة.

يعرض هذا البحث نموذجاً للتحقق من صحة جمل اللغة الإنكليزية من ناحية المعنى، عن طريق توليد قواعد المعنى Semantic Rules من قاعدة بيانات تتضمن الكلمات الأكثر تكراراً في اللغة الإنكليزية، وذلك بالاعتماد على احدى خوارزميات التنقيب في المعطيات وهي خوارزمية FP Growth، التي تقوم بتوليد قواعد الترابط Association Rules بين الكلمات. ولكن هذه القواعد الناتجة عن الخوارزمية تتجاهل تسلسل الكلمات، والذي يعتبر امراً مهماً في تحليل المعنى. لذلك تم اقتراح تعديل على هذه الخوارزمية للحصول على قواعد ترابط تعطي أهمية لتسلسل ورود الكلمات مع مراعاة الزمن اللازم للحصول على هذه القواعد والذاكرة اللازمة لتخزينها.

الكلمات المفتاحية: تحليل المعاني، التنقيب في المعطيات، قواعد المعنى، خوارزمية FP Growth، قواعد الترابط.

[*] أستاذ في جامعة طرطوس – كلية هندسة تكنولوجيا المعلومات والاتصالات – قسم هندسة المعلومات.
[**] طالبة ماجستير جامعة طرطوس – كلية هندسة تكنولوجيا المعلومات والاتصالات – قسم هندسة المعلومات.

مجلة جامعة طرطوس للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (6) العدد(10) 2022
Tartous University Journal for Research and Scientific Studies - engineering Sciences Series Vol. (6) No. (10) 2022

Enhancing FP-growth Algorithm for better Semantic Analysis of English Language Sentences

Prof.Eng. Yaroub Dayoub [*]
Eng. Miriam Obied [**]

(Received 16/8/ 2022 . Accepted 6/11/ 2022)

ABSTRACT

Proofreading the meanings of the natural languages is considered one of the main goals of linguists and those who are interested in computational linguistics. where it is essential to check written texts on computers in different areas.This work presents a model for validating the meaning of English sentences through generating semantic rules from a database which includes the most recurrent words in English language based on one of the data mining algorithms which is FP Growth algorithm.This algorithm generates association rules for the words. However these rules ignore the sequence of words which is considered an important issue in semantic analysis. For this purpose, the algorithm has been modified in this work in order to get association rules which give importance to the sequence of words taking into consideration the time needed to get those rules and the memory needed to store them.

Keywords: Semantic Analisis, Data Mining, Semantic Rules, FP Growth algorithm, Association Rules.

[*] Professor in Information and Communication Technology Engineering Faculty, Information Technology Department.
[**] Master Student in Information and Communication Technology Engineering Faculty, Information Technology Department.

1. مقدمة:

التنقيب في المعطيات Data Mining هو عملية اكتشاف المعلومات الموجودة في مجموعة ضخمة من البيانات [1]. وهي آلية تهدف الى تحليل كميات كبيرة من البيانات لاستخراج نماذج وقواعد مهمة منها لم تكن معروفة مسبقاً والتي لا يمكن اكتشافها بالطرق التقليدية بسبب ضخامة حجم البيانات أو العلاقات المعقدة جداً بين البيانات [2]. ولتطبيق تقنيات التنقيب في المعطيات نحتاج الى:

- توفر قاعدة بيانات ضخمة تتضمن بيانات عن المسألة المراد حلها.
- اختيار وتطبيق خوارزمية تناسب المسألة المطروحة.

العديد من الأبحاث السابقة خاضت تجربة تحليل المعنى في مجالات مختلفة، مثل قواعد البيانات [3]، المنشورات العلمية [4]، المجال الطبي [5] واسترجاع المعلومات [6] وغيرها الكثير من المجالات.

تم استخدام تقنيات مختلفة لتحليل المعنى، مثل تقنية التنقيب في النصوص Text Mining لاستخلاص قواعد الترابط بين الكلمات، حيث قام بعض الباحثون ببناء أنظمة لتحليل المعنى ولكن هذه الأنظمة تتجاهل تسلسل ورود الكلمات في النص وتركز على الكلمات المهمة فقط وعلى توزيعها الاحصائي [7]. استخدم باحثون آخرون تقنيات التنقيب في المعطيات لبناء نموذج لتحليل المعنى يتحقق من معاني جمل اللغة الإنكليزية الصحيحة قواعدياً باستخدام خوارزميات التنقيب في المعطيات التي تهتم بإيجاد قواعد الترابط بين البيانات مثل خوارزميتي Apriori و FP Growth وأبرزوا محاسن وعيوب هاتين الخوارزميتين دون انجاز أي تحسين عليهما [8]. في حين قام باحثون آخرون باستخدام المنطق الضبابي fuzzy logic في إيجاد المعنى المقصود للكلمة في جملة معينة بالاعتماد على قاعدة بيانات معجمية تربط الكلمات من خلال علاقات مختلفة وإعطاء وزن للعلاقات بين الكلمات حسب أهميتها، وتمثيل ذلك في مخطط ضبابي Fuzzy graph [9]. ولجأ آخرون إلى الشبكات العصبونية لبناء نموذج لتحليل جودة النص المكتوب باللغة الإنكليزية بالاعتماد على شبكة عصبونية متكررة Recurrent Neural Network [10]، ولكن ابحاثهم كانت تستند لنتائج تجريبية فقط.

تم في هذا البحث الاعتماد على خوارزمية FP Growth للحصول على قواعد المعنى المناسبة لاستخدامها في التحقق من صحة معاني جمل اللغة الإنكليزية. وتم اقتراح تعديل على خوارزمية FP Growth للحصول على قواعد معنى تعطي أهمية لتسلسل ورود الكلمات مع مراعاة الزمن اللازم للحصول على هذه القواعد والذاكرة اللازمة لتخزينها.

2. أهمية البحث وأهدافه:

تندرج عملية التحقق من معاني جمل اللغة الإنكليزية ضمن مجال معالجة اللغات الطبيعية NLP Natural Language Processing [11] والذي يعد مجالاً هاماً يربط بين علوم الحاسب وعلم اللغة. لازالت عملية فهم نص من قبل الحاسب والحكم عليه فيما إذا كان صحيحاً من ناحية المعنى من المشاكل الكبرى التي تواجه التطبيقات المعلوماتية، كما أنه لا تتوفر حتى الآن أداة برمجية للتحقق من معاني جمل اللغة الإنكليزية.

يهدف هذا البحث إلى تحليل معاني جمل اللغة الإنكليزية والتحقق من صحتها، وذلك بالاعتماد على خوارزمية FP Growth التي تم تطبيقها على قاعدة بيانات تتضمن الكلمات المترافقة باللغة الإنكليزية، للحصول على قواعد المعنى المناسبة، لكن القواعد الناتجة عن هذه الخوارزمية لا تراعي تسلسل ورود الكلمات الذي يعتبر أمراً مهما في

تحليل المعنى. لذلك تم اقتراح تعديل على خوارزمية FP Growth للحصول على قواعد معنى تعطي أهمية لتسلسل ورود الكلمات مع مراعاة الزمن اللازم للحصول على هذه القواعد والذاكرة اللازمة لتخزينها.

تأتي أهمية هذا البحث في انه يقترح آلية لمعالجة اللغات الطبيعية وبناء أنظمة تحاكي الانسان بمجال اللغة، كما أنه يعد خطوة مهمة لتطبيقه في مجالات مختلفة مثل تدقيق رسائل البريد الالكتروني والمقالات والأبحاث العلمية وتطوير عملية التصحيح في الامتحانات الإلكترونية وكما انه يوفر أداة تعليمية مفيدة للغات الطبيعية.

3. طرائق البحث ومواده:

تم اولاً تحديد المسألة المراد حلها وهي إيجاد قواعد المعنى لاستخدامها في التأكد من صحة جمل اللغة الانكليزية من ناحية المعنى، ثم تحديد المتطلبات اللازمة لحل هذه المسألة وهي:

3-1- قاعدة بيانات تتضمن كلمات اللغة الإنكليزية:

تم الحصول على قاعدة بيانات ضخمة تتضمن بيانات عن كلمات اللغة الانكليزية من موقع انترنت [12]، والذي يتضمن ملفات نصية تحتوي كلمات متسلسلة باللغة الإنكليزية مع الصنف الاعرابي لكل كلمة مثل (فعل، اسم، صفة، حرف جر....). هذه الملفات تتضمن حوالي مليون سجل للكلمات المترافقة الأكثر تكراراً في اللغة الإنكليزية لكل من الحالات (كلمتين متسلسلتين، ثلاث كلمات متسلسلة، أربع كلمات متسلسلة، خمس كلمات متسلسلة). يوضح الجدول (1) عينة عشوائية من الملف الذي يحتوي ثلاث كلمات متسلسلة.

الجدول (1): عينة عشوائية من الملف الذي يحتوي ثلاث كلمات متسلسلة

word1	word2	word3	pos1	pos2	Pos3
deep	blue	Sea	jj	jj	nn1
eat	and	Drink	vv0	cc	vv0
went	on	Sale	vvd	ii	nn1

يحتوي الجدول السابق على الاعمدة التالية:

- الاعمدة **word1** و**word2** و**word3** تتضمن الكلمات الثلاثة حسب تسلسل ورودها.
- الاعمدة **pos1** و**pos2** و**pos3** تتضمن الصنف الاعرابي للكلمات الأولى والثانية والثالثة على التوالي.

يمكن توضيح الاختصارات الممثلة للصنف الاعرابي لبعض الكلمات، بالجدول (2) مرتبة حسب الترتيب الابجدي:

الجدول (2): الاختصارات الممثلة للصنف الاعرابي

الرقم	الرمز	التوصيف
1	cc	اداة ربط coordinating conjunction
2	ii	حرف جر عام general preposition
3	jj	صفة عامة general adjective
4	nn1	اسم مفرد singular noun
5	vv0	الشكل الأساسي للفعل base form of lexical verb
7	vvd	الزمن الماضي للفعل past tense of lexical verb

تم اضافة هذه الملفات الى قاعدة بيانات MySQL، وبعد ذلك تم إنشاء قاعدة بيانات تتضمن جداول توافق أربعة عشر مجموعة لقواعد المعنى وبالاعتماد على الملفات السابقة، حيث تم تنظيم الجداول كالتالي:

- الجدول Noun_Verb يخزن الاسماء وما يليها من أفعال.
- الجدول Noun_Adjective يخزن الأسماء وما يليها من صفات.
- الجدول Noun_Noun يخزن الاسماء وما يليها من أسماء.
- الجدول Noun_Preposition يخزن الأسماء وما يليها من أحرف جر.
- الجدول Verb_Verb يخزن الأفعال وما يليها من أفعال.
- الجدول Verb_Adjective يخزن الأفعال وما يليها من صفات.
- الجدول Verb_Noun يخزن الأفعال وما يليها من أسماء.
- الجدول Verb_Preposition يخزن الأفعال وما يليها من أحرف جر.
- الجدول Adjective_Adjective يخزن الصفات وما يليها من صفات.
- الجدول Adjective_Noun يخزن الصفات وما يليها من أسماء.
- الجدول Adjective_Preposition يخزن الصفات وما يليها من أحرف جر.
- الجدول Preposition_Verb يخزن أحرف الجر وما يليها من أفعال.
- الجدول Preposition_Adjective يخزن أحرف الجر وما يليها من صفات.
- الجدول Preposition_Noun يخزن أحرف الجر وما يليها من أسماء.

بما أن الجداول التي تم الحصول عليها من الموقع تتضمن حوالي مليون سجل للكلمات المترافقة والذي يعتبر عدداً كبيراً من السجلات، اي زمن البحث فيها سيكون كبيراً جداً لذلك تم تقسيمها الى أربعة عشر جدول لتشمل اغلب حالات قواعد المعنى، وتم لاحقاً تطبيق خوارزمية FP Growt على هذه الجداول للحصول على قواعد المعنى.

3-2- دراسة الخوارزميات المستخدمة لتنفيذ التحليل المعنوي:

هناك العديد من خوارزميات التنقيب في المعطيات الفعالة والقابلة للتطوير لاستخدمها في إيجاد الترابط بين البيانات، وأشهر تلك الخوارزميات هي Apriori وFP Growth وVertical data format:

تستخدم خوارزمية Apriori طريقة التوليد والاختبار generate and test، وتقوم بمسح متكرر لقاعدة البيانات، أي انها تولد مجموعة من العناصر(الكلمات) المرشحة ثم تختبر فيما إذا كانت تمثل عناصر متكررة، بالتالي فإن هذه الخوارزمية تستغرق زمناً كبيراً بالإضافة الى مساحة تخزينية كبيرة ناتجة عن استراتيجية البحث Breadth First Search، خاصة إذا كانت قاعدة البيانات ضخمة وكان عدد العناصر المرشحة المختبرة كبيراً [13].

تعتمد خوارزمية Vertical data format [1] على مبدأ تحويل الاسطر في الجدول الى أعمدة والمسح المتكرر لقاعدة البيانات، أي أنه من أجل كل عنصر في قاعدة البيانات يتم تخزين قائمة بأرقام الاسطر التي يتواجد فيها العنصر، وبذلك يتم تمثيل البيانات بشكل عمودي، وبعد ذلك يتم حساب مجموعات العناصر المتكررة، تعتبر هذه الخوارزمية سريعة بسبب استراتيجية البحث (Depth First Search) لكن القوائم الوسيطية التي تضم أرقام أسطر العناصر قد تكون ضخمة جداً خاصة إذا كانت قاعدة البيانات كبيرة [14]، وبالتالي تتطلب زمن تنفيذ ومساحة تخزينية أكبر، مما يحد من استخدامها.

يقوم المبدأ الأساسي لخوارزمية (Frequent Pattern Growth (FP Growth [1] على استراتيجية فرق تسد divide and conquer، التي تقوم بمسح قاعدة البيانات مرتين فقط فتحول قاعدة البيانات الضخمة الى بنية شجرية حجمها أصغر من حجم قاعدة البيانات الاصلية، تسمى شجرة النموذج المتكرر (FP frequent pattern tree(tree، وتعتبر تطويراً هاماً لخوارزمية Apriori لأنها تعتمد على التنقيب في الشجرة للحصول على العناصر المتكررة دون توليد عناصر مرشحة، واذا كانت قاعدة البيانات كبيرة فإن بناء شجرة FP tree قد يستغرق وقتاً، لكن بمجرد بنائها تتم قراءة العناصر المتكررة بسهولة [8] [15].

بعد دراسة ومقارنة افضل خوارزميات التنقيب في المعطيات التي تختص بإيجاد الترابط بين البيانات، تم الاعتماد على خوارزمية FP Growth لتوليد قواعد المعنى في هذا البحث، علماً انه تم تنفيذ الخوارزمية بلغة الجافا، وهي لغة برمجية غرضية التوجه Object Oriented Programming تسمح ببناء تطبيقات برمجية تعمل على أنظمة تشغيل مختلفة، وتم العمل على بيئة تطويرية مناسبة لهذه اللغة هي برنامج NetBeans IDE 8.2 وهو تطبيق برمجي يوفر تسهيلات شاملة لتطوير البرمجيات بلغة الجافا بالإضافة الى لغات اخر، ويحتوي على عدة أدوات مساعدة تسمح للمطورين بإنشاء تطبيقات برمجية باستخدام واجهة المستخدم الرسومية، وتم التنفيذ على حاسب بالمواصفات التالية: المعالج intel core i7-2.20GHz والذاكرة 8GB ونظام Windows 7.

4. الآلية المقترحة للتحقق من صحة الجملة:

قبل التحقق من صحة الجملة من ناحية المعنى يجب التحقق أولاً من صحتها مفرداتياً ثم قواعدياً، وليس بالضرورة أن تكون كل جملة صحيحة قواعدياً هي جملة صحيحة من ناحية المعنى، فمثلاً الجملة The car eats the apple هي جملة صحيحة قواعدياً ولكنها غير صحيحة من ناحية المعنى.

يتم التحقق من صحة الجملة مفرداتياً باستخدام قاموس (جدول) يتضمن كلمات اللغة الإنكليزية مع نوع كل كلمة (فعل، اسم، صفة، حرف جر....) وذلك للتأكد من أن مفردات(كلمات) الجملة هي ضمن كلمات اللغة الإنكليزية، والجدول التالي يمثل عينة عشوائية من القاموس المستخدم:

الجدول (3): عينة من القاموس المستخدم

number	Word	Type
1	A	Determiner
2	Abandon	Verb
3	Abandoned	Adjective
4	Abandonment	Noun

ثم يتم التحقق من صحة الجملة نحوياً باستخدام النحو الشكلي Formal Grammar، والذي هو مجموعة من الأسس والقواعد التي تبين طريقة تكوين عبارات اللغة البسيطة والمركبة [16]، ويرمز له بالشكل $G=(V_N,V_T,P,S)$ حيث:

V : مجموعة مفردات اللغة والتي يمكن توليدها باستخدام النحو الشكلي G وهي نوعان:

V_N: non_terminal symbols مجموعة العناصر الثانوية وتمثل العقد الداخلية في شجرة النحو.

V_T: terminal symbols مجموعة العناصر النهائية وتمثل الأوراق في شجرة النحو.

$V_T \cup V_N = V$ و $V_T \cap V_N = \Phi$ حيـــــث
P: productions rules مجموعـة قواعـد الاشـتقاق ولهـا الصـيغة العامـة $A \rightarrow \alpha$ β حيث $\alpha \in V^*$.
S: start symbol محرف (عنصر) البداية ويمثل جذر شجرة النحو، حيث $S \in V_N$.

وهناك عدة أنواع للنحو الشكلي، تم استخدام النحو الحر context –free formal grammar [17]، حيث ان اللغات المولدة عن هذا النحو تدعى باللغات type2 او context-free langue لأنها لا تتعلق بحيز ما، فلا توجد قيود على نوعية المحارف الواقعة على يمين السهم في قواعد الاشتقاق المولدة بالتالي يمكن توليد لغات عملاقة ذات مزايا واسعة جدا كافية لتوصيف وتحليل أعقد المسائل العلمية واللغات الخاصة بالتعرف على الصور والصوت والاشكال وغيرها، وقد تم بناء نموذج نحوي للتحقق من صحة الجملة نحوياً، فإذا كانت سلسلة مفردات الجملة متطابقة مع احدى قواعد النموذج النحوي تكون الجملة صحيحة نحوياً والا تعتبر غير صحيحة نحوياً، والقواعد التالية تمثل جزء من النموذج النحوي المستخدم:

<S> → <NP> <VP> | <NPP> <VP> |<VP> | ……
<NP> → <N> | <Det> <Adj> <N> | <Det> <N> |<Pron> | <Pron> <N> | ……
<VP> → <V> <NP> |<V> <VPP> <NP> |<V> <NPP> <NP> | ……
<NPP> → <Prep> <NP>
<AP> → <Adj> | <Adj> <Adj> | <Adj> <Conj> <Adj>
<APP> → <Prep> <AP>
<V> → <V> |<V> <V> |<V> <Adv> <V> |<V> <Neg> <V> |……
<VPP> → <Prep> <V>

تم توضيح بعض الرموز المستخدمة في النموذج النحوي السابق، في الجدول (4) التالي:

الجدول (4): الاختصارات المستخدمة في النموذج النحوي

الاختصارات	المعنى
S	Sentence
Det	Determiner
Adj	Adjective
Pron	Pronoun
Conj	Conjunction
Neg	Negation
Prep	Preposition
Adv	Adverb
V	Verb
N	Noun
NP	Noun Phrase
VP	Verb Phrase
AP	Adjective Phrase
NPP	Noun Preposition Phrase
VPP	Verb Preposition Phrase
APP	Adjective Preposition Phrase

بعد التحقق من صحة الجملة مفرداتياً ونحوياً يتم استخدام قواعد المعنى الناتجة عن خوارزمية FP Growth في التحقق من صحة معنى الجملة، حيث أن قواعد المعنى سيكون لها الشكل التالي:

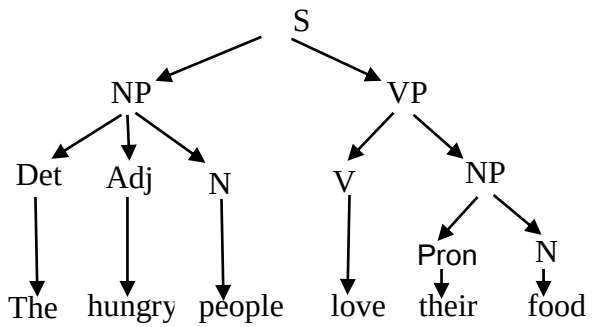
<Sentence> → <part><part>
<part> → <Noun> | <Verb> | <Adjective> | <Preposition> | <Sentence>

المثال التالي يوضح الآلية المقترحة للتحقق من صحة الجملة، لتكن لدينا الجملة التالية: The hungry people love their food حيث يتم التحقق من صحة الجملة وفق الخطوات التالية:

1- التحقق اولاً من صحة الجملة مفرداتياً وتحديد الصنف الاعرابي لكل كلمة، عن طريق قاموس كلمات اللغة الإنكليزية الموضح في الجدول (3):

The **(Determiner)** hungry **(adjective)** people **(noun)** love **(verb)** their **(Pronoun)**
food **(noun)**

2- التحقق من صحة الجملة نحوياً عن طريق النموذج النحوي السابق، وبناء شجرة النحو للجملة كما يلي:



الشكل(1): شجرة النحو للمثال

3- حذف الكلمات غير المهمة من ناحية المعنى:

مثل أدوات الربط بين أجزاء العبارة وأدوات التعريف والضمائر الشخصية، في مثالنا هذا تم حذف أداة التعريف the والضمير their، فتأخذ العبارة الشكل الآتي:

hungry people love food

4- مناقشة جميع حالات الكلمات المترابطة:

• التحقق من وجود الكلمتين hungry people ضمن قواعد المعنى الخاصة بالصفات وما يليها من أسماء.

• التحقق من وجود الكلمتين people love ضمن قواعد المعنى الخاصة بالأسماء وما يليها من افعال.

• التحقق من وجود الكلمتين love food ضمن قواعد المعنى الخاصة بالأفعال وما يليها من اسماء.

فعند إيجاد الكلمات السابقة جميعها ضمن قواعد المعنى، عند ذلك يتم التأكد من صحة الجملة من ناحية المعنى.

5. مبدأ عمل خوارزمية FP Growth:

قبل شرح مبدأ عمل الخوارزمية سنوضح بعض المفاهيم الأساسية [18]:

• **الدعم الأصغري (min_sup) minimum_ support**: هو ثابت عددي موجب، أكبر او يساوي الواحد، يتم تحديده من قبل المستخدم، حيث يعرّف الدعم لعنصر ما بأنه عدد الأسطر في قاعدة البيانات التي تتضمن هذا العنصر.

•**العنصر المتكرر frequent item**: نقول عن عنصر أنه عنصر متكرر إذا وفقط إذا كان تكراره أكبر أو يساوي الدعم الأصغري min_sup، لتكن $I=\{a_1, a_2, \dots a_m\}$ مجموعة من العناصر، و $DB=\{T_1,T_2,\dots, T_n\}$ مجموعة تمثل اسطر قاعدة البيانات، حيث $T_i\ (i \in [1\text{-}n])$ هو سطر في قاعدة البيانات يحتوي على مجموعة من I.

• **شجرة النموذج المتكرر frequent pattern tree (FP tree)**: هي بنية شجرية مكونة من جذر واحد يسمى root، ومجموعة من الأشجار الفرعية الأبناء للجذر التي تتكون من عقد والروابط بين العقد، حيث كل عقدة تتكون من اسم العنصر الذي تمثله العقدة، وعداد يمثل عدد مرات تكرار القعدة.

دخـــــــل الخوارزميـــــــــة: DB قاعـــــدة البيانـــــات، min_sup الـــــدعم الاصـــــغري.

خرجهـــــــــــــــا: قواعـــــــــد التـــــــــرابط (قواعـــــــــد المعنـــــــــى).

خطوات عمل الخوارزمية [1]:

1. المرور الأول على قاعدة البيانات لإيجاد itemsets -1 مجموعة تتضمن كل عنصر مع تكراره وتخزينها في جدول (H-table) Header-table.
2. ايجاد العناصر المتكررة frequent items التي تكرارها أكبر او يساوي الدعم الاصغري.
3. ترتيب العناصر المتكررة في الجدول H-table ترتيباً تنازلياً بحسب قيمة تكرارها.
4. المرور الثاني على قاعدة البيانات لبناء شجرة النموذج المتكرر FP tree كما يلي:
   a. انشاء جذر الشجرة وليكن root.
   b. ترتيب العناصر المتكررة في كل سطر من أسطر قاعدة البيانات بحسب الترتيب في الجدول H-table، أي بحسب قيمة تكرارها تنازلياً، ولتكن قائمة العناصر المتكررة في السطر T هي [P|S] حيث P هو العنصر الأول و S هو القائمة المتبقية.
   c. انشاء فرع في الشجرة لكل سطر من أسطر قاعدة البيانات، كما يلي: إذا كان لدى الشجرة ابن N، بحيث يكون N.item-name = P.item-name, يتم زيادة عداد العقدة N بمقدار واحد، والا يتم انشاء عقدة جديدة N يكون العداد فيها مساوياً الواحد مع ربطها مع العقدة الاب لها، ويتم تكرار الآلية السابقة حتى تصبح القائمة S فارغة.
5. التنقيب في الشجرة للحصول على قواعد الترابط: يتم التنقيب في الشجرة عن طريق تحديد المسارات المرتبطة في شجرة FP tree لكل عقدة وهو ما يسمى قاعدة النموذج الشرطي Conditional Pattern Base، حيث لكل نموذج بطول 1 (نموذج لاحق ابتدائي prefix pattern) يتم بناء قاعدة النموذج الشرطي، وبالاعتماد على المسارات التي تم ايجادها في قاعدة النموذج الشرطي يتم بناء شجرة FP الشرطية Conditional FP

(tree) لكل عقدة، حيث يتم أخذ العناصر المتكررة فقط (التي تكرارها اكبر او يساوي الدعم الاصغري)، ومن ثم يتم الحصول على قواعد الترابط من خلال النماذج المتكررة المولدة من شجرة FP الشرطية.

5-1- مثال عملي على خوارزمية FP Growth :

تم توضيح مبدأ عمل خوارزمية FP Growth من خلال المثال البسيط التالي، حيث تم تطبيق الخوارزمية على الجدول (5) الذي يتضمن ثمانية أسطر لتتالي فعل واسم، وذلك من أجل دعم اصغري min_sup=1.

الجدول (5): البيانات المخزنة

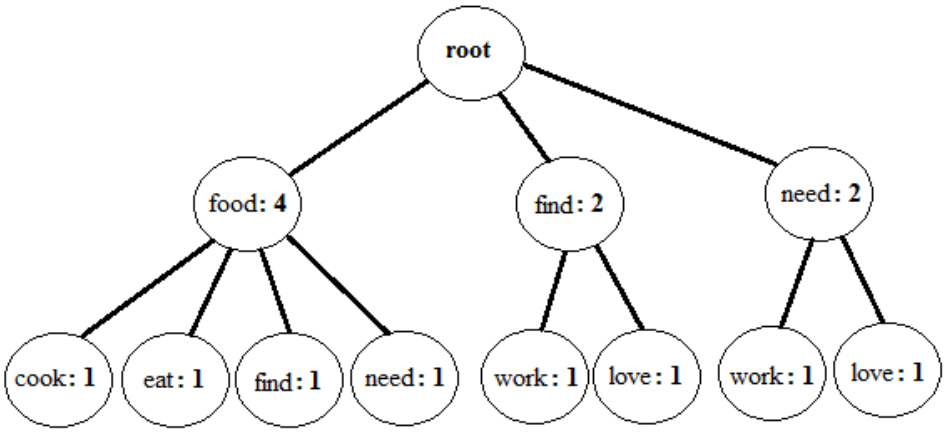
ID	Verb	Noun
T1	Cook	Food
T2	Eat	Food
T3	Find	Food
T4	Find	Work
T5	Find	Love
T6	need	Food
T7	need	Work
T8	need	Love

عند المرور الأول على الجدول (5) يتم تشكيل جدول H-table (الجدول (6))، وهو مجموعة من العناصر المتكررة مع تكرارها بعد ترتيب العناصر المتكررة ترتيباً تنازلياً بحسب قيمة تكرارها.

الجدول (6): جدول العناصر المتكررة H-table عند تطبيق خوارزمية FP Growth

Item_id	Support
food	4
find	3
need	3
work	2
love	2
cook	1
eat	1

عند المرور الثاني على الجدول (5) يتم بناء شجرة العناصر المتكررة FP tree الموضحة بالشكل(2):



الشكل(2): شجرة FP tree بعد تطبيق خوارزمية FP Growth

أخيراً يتم التنقيب في شجرة FP tree كما يلي:

من أجل العقدة النهائية (الورقة) work، تم إيجاد مسارين لها ضمن الشجرة هما {need:1} و{find:1} وهذان المساران يشكلان قاعدة النموذج الشرطي للعقدة work، ثم يتم بناء شجرة FP tree الشرطية للعقدة work بناءً على المسارين السابقين (لم يتم حذف أي مسار عند بناء شجرة FP tree الشرطية لأن الدعم الأصغري يساوي الواحد)، من خلال المسارات في شجرة FP tree الشرطية يتم توليد مجموعة النماذج المتكررة وهي{need,work} و{find,work}، وبنفس الطريقة يتم التنقيب في شجرة FP tree بالنسبة لبقية العقد كما هو موضح في الجدول(7):

الجدول (7): التنقيب في شجرة FP tree

item	Conditional pattern base	Conditional FP tree	Frequent patterns generated
work	{need:1}{find:1}	need:1,find:1	{need,work}{find,work}
love	{need:1}{find:1}	need:1,find:1	{need,love }{find,love}
need	{food:1}	food:1	{ food, need }
find	{food:1}	food:1	{ food, find }
eat	{food:1}	food:1	{ food, eat }
cook	{food:1}	food:1	{ food, cook }

من خلال التنقيب في الشجرة نحصل على قواعد الترابط التالية:

if word1='food' **then** word2='cook'
if word1='food' **then** word2='eat'
if word1='food' **then** word2='find'
if word1='food' **then** word2='need'
if word1='need' **then** word2='work'
if word1='need' **then** word2='love'
if word1='find' **then** word2='work'
if word1='find' **then** word2='love'

نلاحظ أن بعض هذه القواعد تعتبر قواعد معنوية خاطئة ومعكوسة، مما يؤدي لزيادة زمن التحليل المعنوي وهدر بحجم الذاكرة، فقواعد الترابط السابقة أظهرت الترابط بين الكلمات، لكنها تجاهلت تسلسل ورود الكلمات، لأن خوارزمية FP growth تركز على ترتيب العناصر المتكررة ترتيباً تنازلياً بحسب قيمة تكرارها، هذا يعني أنه كلما كانت الكلمة أكثر تكراراً كلما كانت أقرب إلى جذر الشجرة، لكن موضوع ترتيب الكلمات يعتبر أساسيا في هذا البحث لأنه يؤثر على إعطاء المعنى الصحيح للجملة، لذا اصبح من الضروري تعديل الخوارزمية للحصول على قواعد ترابط تعطي أهمية لتسلسل ورود الكلمات.

6. الخوارزمية المقترحة:

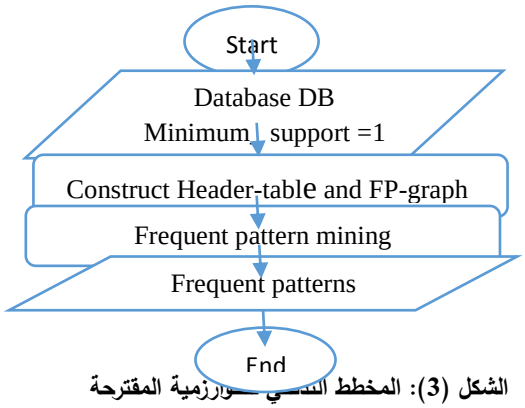
التعديل على خوارزمية FP growth كان من خلال حذف الخطوة (ترتيب العناصر المتكررة ترتيباً تنازلياً بحسب قيمة تكرارها)، وفرض قيد على قيمة الدعم الأصغري بحيث تكون قيمته تساوي الواحد، والهدف من هذا القيد هو عدم حذف قواعد الترابط التي تتضمن كلمات يكون تكرارها أقل من قيمة الدعم الأصغري، لأن حذف بعض قواعد الترابط يؤدي الى اعتبار الجمل التي تتضمن واحداً او أكثر من القواعد المحذوفة غير صحيحة من ناحية المعنى على الرغم من وجودها ضمن قاعدة البيانات، كما ان الدعم الاصغري هو قيمة تحدد من قبل المستخدم تجريبياً، مما يجعل النتائج محفوفة بالأخطاء، وقد لاحظنا انه يتم بناء شجرة FP tree بطريقة تجعل عقد المستوى الأول في الشجرة لا تتكرر، بينما يوجد احتمال لتكرار العقد في المستوى الثاني وما بعده، لذلك تم اقتراح تحسين على الخوارزمية لتقليل عدد العقد، وهو استبدال بنية الشجرة بالمخطط graph، حيث يعرّف المخطط G(V,E) على انه بنية بيانات غير خطية

تتكون من مجموعة من العقد المتصلة بواسطة خطوط ارتباط، حيث Vertices:V هي مجموعة العقد ، و Edges:E هي مجموعة خطوط الارتباط (الحواف) بين العقد [19]، وهناك أنواع مختلفة للمخططات، لكن النوع الذي تم استخدامه في الخوارزمية المقترحة هو DAG (directed acyclic graph) مخطط موجه بدون حلقات، فتصبح خطوات تنفيذ الخوارزمية المقترحة كالآتي:

1.المرور على قاعدة البيانات لبناء جدول Header-table الذي يتضمن العناصر المتكررة فقط دون ترتيبها او ذكر تكرارها وبناء المخطط FP-graph.

2.التنقيب في المخطط FP-graph للحصول على قواعد الترابط.

والشكل التالي يمثل المخطط التدفقي للخوارزمية المقترحة:



الشكل (3): المخطط التدفقي للخوارزمية المقترحة

سيتم شرح هذه الخطوات بالتفصيل في الفقرات التالية، لكن اولاً سنقوم بتوضيح بعض المفاهيم المستخدمة في الخوارزمية المقترحة، حيث يتألف كل سطر في جدول Header-table من حقلين الأول مُعرف العنصر itemId والثاني مؤشر لعقدة هذا العنصر في المخطط itemNodePointer، لم يتم ذكر عدد مرات تكرار كل عقدة، لأنه في موضوع تحليل المعاني لا يهم عدد مرات تكرار كل كلمة، وهذا يقلل من المساحة التخزينية اللازمة للتخزين، أما بالنسبة للمخطط يحوي عدد عقد مساوي للعناصر المتكررة لأول مجموعة (1-itemsets)، وكل عقدة في المخطط تتألف من حقلين، الأول مُعرف العنصر، والثاني عبارة عن قائمة Parent-list والتي تضم مؤشر آباء هذه العقدة ParentNodePointer، وكل حافة E_{ij} بين عقدتين V_i و V_j تمثل جزء من سطر في قاعدة البيانات ومن أجل تخزين هذا الجزء عند وروده في أكثر من سطر تم تزويد كل حافة بقائمة TransId-list وذلك لتخزين الأسطر المشتركة بتلك الحافة، بالتالي كل الأسطر في قاعدة البيانات يتم تمثيلها من خلال Parent-list والتي تحوي مؤشرات للعقد الآباء لهذه العقدة ParentNodePointer بالإضافة إلى TransId-list التي تحوي أرقام الأسطر والتي تستخدم من أجل تعليم (Tagging) الحواف بين عقد المخطط.

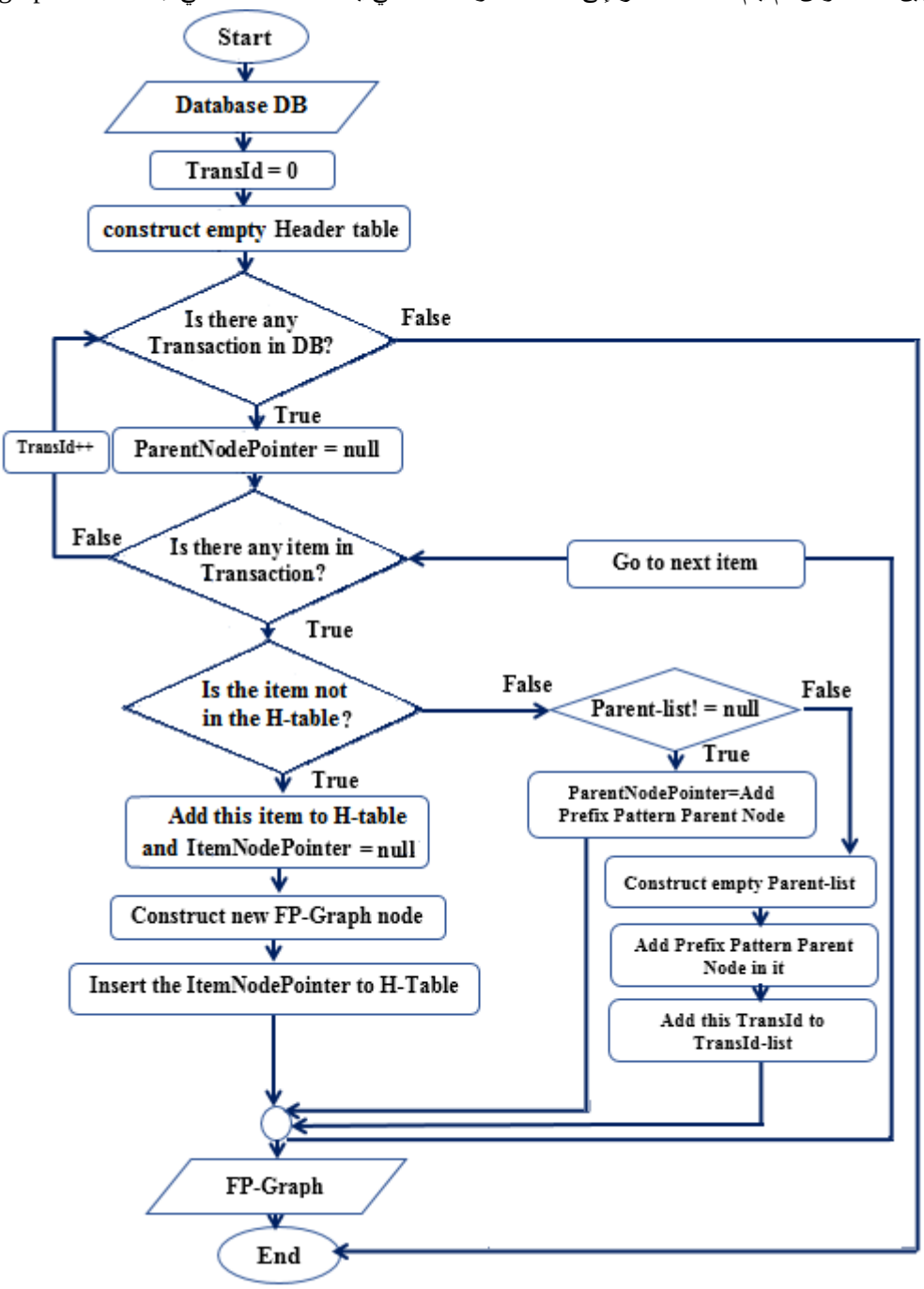
وبما أن المخطط لا يحتوي عقدة جذر root يتم العبور عبر العقد من خلال جدول Header-table، اما البحث عن النموذج اللاحق Prefix Pattern يتطلب فقط معرفة العقد الآباء لعقدة هذا النموذج والتي يتم تخزينها في اللائحة Parent-list، لذلك لا يتم تخزين أبناء العقد في الخوارزمية المقترحة مما يحقق توفير بالزمن والمساحة.

1-6 بناء المخطط FP-graph:

تُقسم عملية بناء المخطط إلى مرحلتين أساسيتين:

1. تهيئة جدول Header-table وعقد المخطط وربطها مع المؤشر itemNodePointer.
2. حفظ أسطر قاعدة البيانات في المخطط.

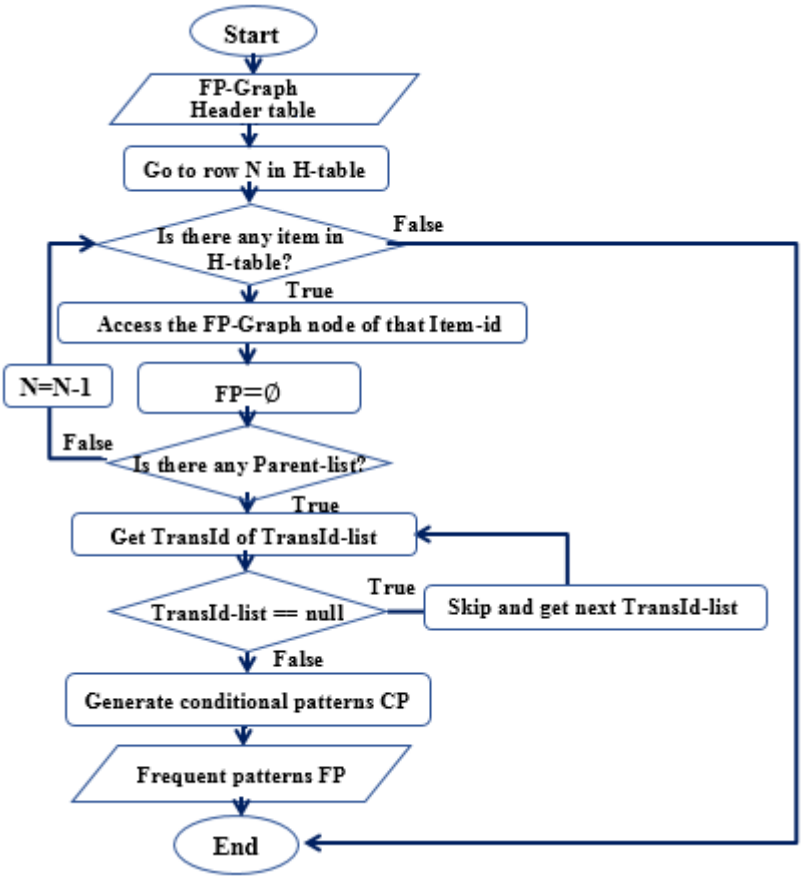
حيث يُشكل المخطط بنية مضغوطة وفعّالة لتخزين العناصر المتكررة لأول مجموعة 1- itemsets حيث يتم تمثيل كل عنصر بعقدة واحدة فقط، وحفظ جميع الأجزاء المتماثلة بين الاسطر في نفس الحافة، حيث يُستخدم رقم السطر من أجل تعليم الحواف بين العقد، ومن ثم يتم ادخال السطر إلى المخطط، والشكل التالي يمثل المخطط التدفقي لبناء مخطط FP-graph :



الشكل (4): المخطط التدفقي لبناء المخطط FP-graph

2-6 التنقيب عن قواعد الترابط في المخطط FP-graph:

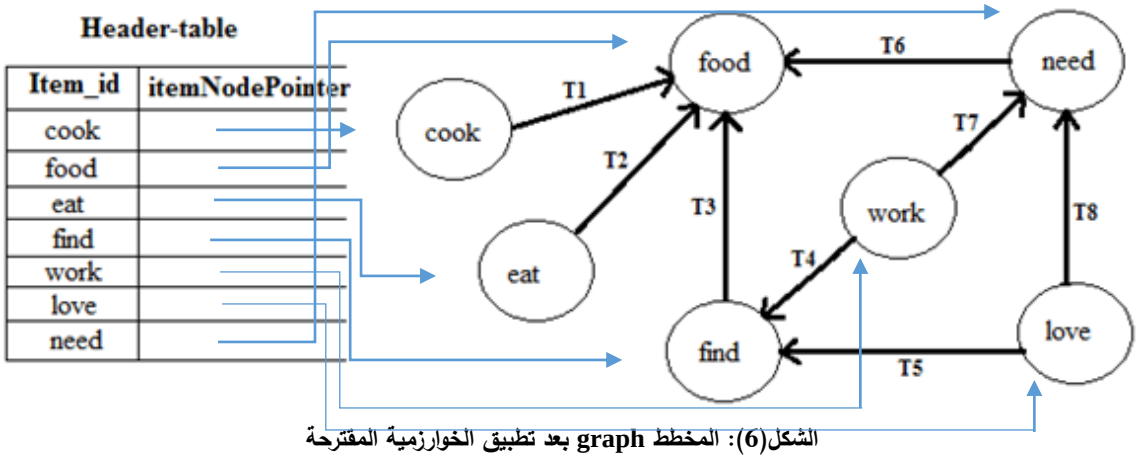
يتم الحصول على قواعد الترابط ممثلة بقواعد النموذج الشرطي Conditional Pattern Base دون بناء أشجار FP الشرطية (كما في خوارزمية FP-growth) مما يؤدي إلى زيادة سرعة الخوارزمية المقترحة، لأنه طالما تم فرض قيد على قيمة الدعم الأصغري أنه يساوي الواحد لا جدوى من بناء الأشجار الشرطية، بل إنه يسبب هدر في الزمن والذاكرة، فلإيجاد النموذج المتكرر يتم استخدام خوارزمية البحث من أسفل جدول Header-table إلى الأعلى من أجل عبور كل عقد المخطط، حيث يتم الانطلاق من العقدة التي تمثل هذا النموذج، ثم يتم الانتقال إلى آباء هذه العقدة حتى نصل إلى عقدة مؤشر الأب ParentNodePointer لها يساوي Null، حيث يتم الوصول الى ParentNodePointer من القائمة Parent-list لكل عقد المخطط، ثم يتم استرجاع قيم TransId من القائمة TransId-list من أجل كل ParentNodePointer لتوليد النماذج الشرطية التي تمثل النماذج المتكررة (قواعد الترابط) المطلوبة، والشكل التالي يمثل المخطط التدفقي للتنقيب عن النماذج المتكررة Frequent patterns، حيث N تمثل العدد الكلي لأسطر جدول Header-table.



الشكل (5): المخطط التدفقي للتنقيب عن قواعد الترابط

3-6 مثال عملي على الخوارزمية المقترحة:

تم تطبيق الخوارزمية المقترحة من أجل المثال السابق نفسه في الجدول (5)، بالتالي تم المرور على الجدول (5) مباشرة لبناء جدول Header-table وبناء المخطط الموضحين بالشكل(6) كما يلي:



الشكل(6): المخطط graph بعد تطبيق الخوارزمية المقترحة

عند المرور على السطر الأول في قاعدة البيانات (cook,food) ذو المعرف T1 يتم إضافة العقدتين cook وfood الى جدول Header-table وانشاء عقدتين cook وfood في المخطط وإضافة مؤشر itemNodePointer لهاتين العقدتين في جدول Header-table ويتم تخزين هذا السطر ضمن المخطط، حيث أن العقدة cook تشير الى العقدة الاب food، بالتالي مؤشر ParentNodePointer للعقدة cook هو food ويتم تخزينه في القائمة TransId-list، ويتم تعليم الحافة بينهما عن طريق إضافة رقم السطر T1 الى القائمة TransId-list، أما مؤشر ParentNodePointer الخاص بالعقدة food يساوي null، وهكذا بالنسبة لبقية أسطر قاعدة البيانات حتى يتم بناء المخطط بالكامل، وبعد بناء المخطط يتم التنقيب عن العناصر المتكررة حيث تبدأ عملية التنقيب من العنصر الموجود أسفل جدول Header-table، بالتالي يتم الانطلاق من سطر العنصر need إلى المخطط FP-Graph من أجل إيجاد الأسطر التي تحوي العنصر need دون الحاجة لمسح قاعدة البيانات، تمتلك العقدة need مؤشر ParentNodePointer يؤشر الى العقدة food بقيمة حافة T6، حيث من اجل السطر T6 يتم الانتقال من العقدة need الى العقدة food وبالتالي النموذج الشرطي الخاص بالعقدة need هو {need,food} ثم يتم الانتقال الى سطر العنصر love في جدول Header-table حيث تمتلك العقدة love مؤشرين ParentNodePointer يؤشران الى العقدتين need وfind بقيمة حافة T8 وT5، حيث من اجل السطر T5 يتم الانتقال من العقدة find الى العقدة love ومن اجل السطر T8 يتم الانتقال من العقدة need الى العقدة love وبالتالي النماذج الشرطية الخاصة بالعقدة love هي {need,love}،{find,love} دون حذف أي نموذج لأن قيمة الدعم الاصغري min_sup=1 وهذه النماذج الشرطية تمثل قواعد الترابط المطلوبة، وبنفس الطريقة يتم استنتاج قواعد الترابط لباقي العناصر موضحة كالتالي:

if word1='cook' **then** word2='food'
if word1='eat' **then** word2='food'
if word1='find" **then** word2='food'
if word1='need' **then** word2='food'
if word1='need' **then** word2='work'
if word1='need' **then** word2='love'
if word1='find' **then** word2='work'
if word1='find' **then** word2='love'

كما هو ملاحظ أن قواعد الترابط السابقة تعتبر قواعد معنوية صحيحة وقد أعطت أهمية لتسلسل ورود الكلمات، كما نلاحظ عدم ورود تكرار لأي عقدة (كلمة) في المخطط، ليصبح عدد العقد في المخطط 7 عقد بدلاً من 12 عقدة كما في خوارزمية FP-growth وبذلك قد تم تقليل عدد العقد بنسبة 41% في هذا المثال.

4-6 دراسة فعالية الخوارزمية المقترحة:

إن نسبة كبيرة من زمن تنفيذ خوارزمية FP-growth الأساسية يستهلك في بناء شجرة FP-Tree والتنقيب في الأشجار الشرطية، وقد تم تقليص هذا الزمن في الخوارزمية المقترحة بالاعتماد على المخطط الموجه كما تم تقليل المساحة التخزينية المطلوبة، وسنوضح ذلك رياضياً لبرهان فعالية الخوارزمية المقترحة ومقارنتها بخوارزمية Growth FP، فمن ناحية **المساحة التخزينية** كل عقدة في المخطط تحتاج الى 2 بايت من أجل معرّف العنصر itemId و4 بايت من أجل Parent-list و4 بايت من أجل TransId-list، بالتالي المساحة التخزينية المطلوبة لتخزين عقدة في المخطط موضحة بالعلاقة (1):

$$Space = 2 + 4N + 4\ N = 2 + 8N \qquad (1)$$

حيث N تمثل عدد العقد الآباء لهذه العقدة، وكل سطر في جدول H-table يتطلب 2 بايت لمُعرف العنصر itemId، و4 بايت لمؤشر عقدة هذا العنصر في المخطط itemNodePointer، بالتالي المساحة التخزينة المطلوبة لكل سطر في جدول H-table هي 6 بايت، الآن سنقوم بحساب المساحة التخزينية المطلوبة للمثال السابق لتوضيح فعالية الخوارزمية المقترحة ومدى ضغطها لقاعدة البيانات، حيث العدد الكلي لعقد المخطط هو 7 عقد فتكون المساحة التخزينية المطلوبة على الشكل التالي:

المساحة المطلوبة لجدول H-table هي: 6*7=42 بايت

المساحة المطلوبة للمخطط: كل عقدة تتطلب مساحة تخزينية تختلف باختلاف عدد العقد الآباء لتلك العقدة، فمثلاً العقدة work تحوي مؤشرين ParentNodePointer في اللائحة Parent-list الأول يشير إلى العقدة need والثاني يشير الى العقدة find، وبالتالي فإن N=2 بالنسبة للعقدة work، والمساحة المطلوبة لهذه العقدة هي 2+(2*8)=18 بايت وفق العلاقة (1) ، وبنفس الطريقة يتم حساب المساحة لبقية العقد، فتكون المساحة المطلوبة لتخزين جميع العقد 78 بايت، بالتالي المساحة الكلية المطلوبة لتخزين المخطط 78+42=120 بايت، مقابل 232 بايت [20] المساحة التخزينية المطلوبة لتخزين شجرة FP-Tree في خوارزمية FP-growth، أي توفير بمقدار 48% من حيث عدد العقد والمساحة التخزينية. أما من ناحية **الزمن**، يعتمد الزمن الذي تستغرقه الخوارزمية المقترحة على ثلاثة أزمنة رئيسية كما هو موضح في العلاقة (2):

$$Time = n * T_1 + m * T_2 + r * T_3 \qquad (2)$$

حيث T_1 يمثل الزمن اللازم لتوليد سطر واحد في جدول H-table، و n: العدد الكلي لأسطر جدول H-table

T_2 يمثل الزمن اللازم لتوليد عقدة واحدة ضمن المخطط، و m: العدد الكلي للعقد، T_3 يمثل الزمن اللازم للتنقيب عن النماذج المتكررة، وهو الزمن المطلوب لقراءة عقدة واقعة في مسار النموذج المتكرر الخاص بكل عنصر موجود في جدول H-table، حيث r: عدد العقد المقروءة. الآن سنقوم بحساب زمن تنفيذ الخوارزمية المقترحة للمثال السابق وفق العلاقة (2)، الزمن اللازم لبناء جدول H-table يساوي ($7*T_1$)، أما الزمن الكلي لبناء المخطط يساوي ($7*T_2$)، ومن أجل التنقيب عن مجموعة العناصر المتكررة الخاصة بالعقدة

work في الخوارزمية المقترحة سيتم قراءة 3 عقد بالتالي الزمن اللازم لإيجادها $3*T_3$ ومن اجل التنقيب عن العناصر المتكررة للعقدة eat سيتم قراءة عقدتين بالتالي الزمن اللازم لإيجادها $2*T_3$ وبنفس الطريقة يتم حساب الزمن اللازم للتنقيب عن بقية عقد المخطط، فيكون زمن التنقيب عن جميع العقد هو $15T_3$، وبالتالي الزمن الكلي لتنفيذ الخوارزمية المقترحة بالنسبة للمثال السابق هو $7*T_1+7*T_2+15*T_3$ أما زمن تنفيذ خوارزمية FP growth [20] هو $7*T_1+12*T_2+30*T_3$ بفرض التنفيذ على نفس الجهاز، بالمقارنة بين زمن تنفيذ الخوارزميتان نجد أن الخوارزمية المقترحة أسرع من خوارزمية FP-growth.

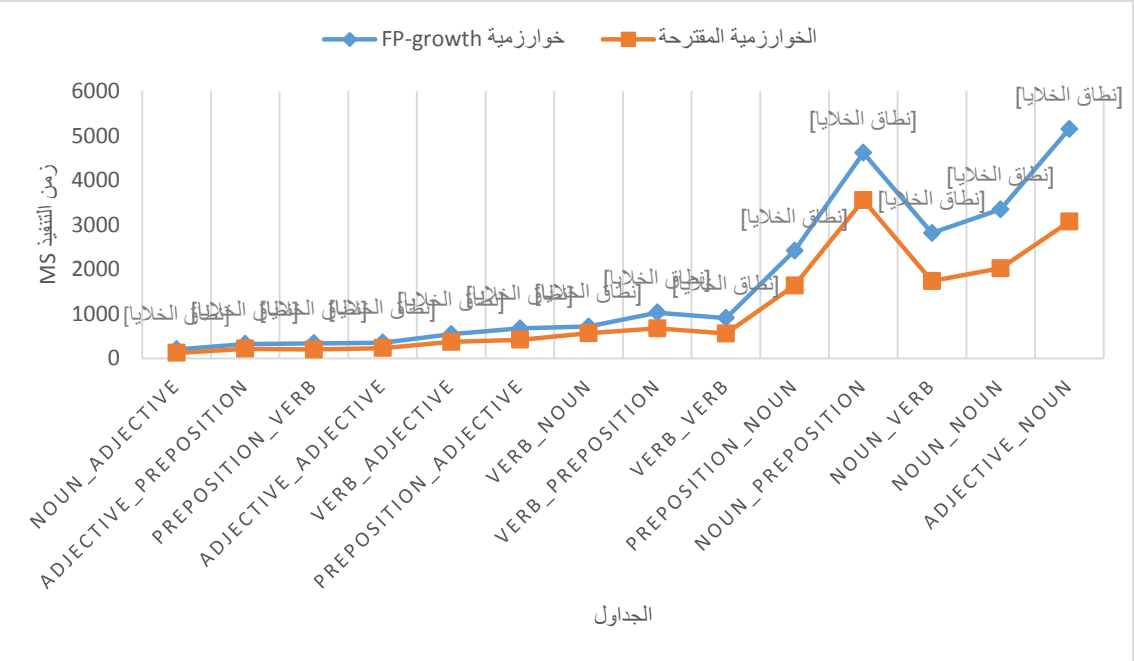
7. النتائج والمناقشة:

تم تطبيق خوارزمية FP Growth والخوارزمية المقترحة المعتمدة على المخطط على الجداول الأربعة عشر لقواعد المعنى التي تم شرحها في الفقرة (3−1)، علماً انه تم تنفيذ الخوارزميتين بلغة الجافا، والجدول التالي يبين عدد العقد ومتوسط زمن تنفيذ الخوارزميتين عند تطبيقهما على الجداول الأربعة عشر مرتبة تصاعدياً حسب عدد الاسطر، بالإضافة إلى النسبة المئوية للتحسين في كل من عدد العقد ومتوسط زمن تنفيذ الخوارزمية:

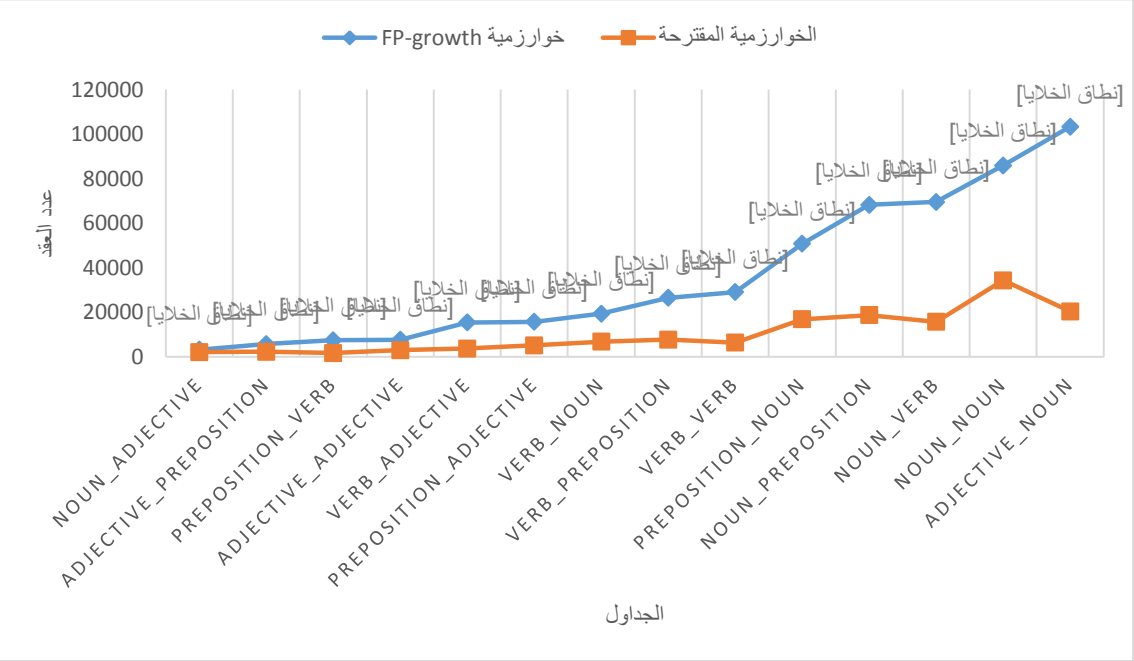
الجدول (8): نتائج تنفيذ خوارزمية FP-Growth والخوارزمية المقترحة

الجدول	عدد الاسطر	خوارزمية FP-Growth		الخوارزمية المقترحة		النسب المئوية للتحسين	
		عدد العقد	زمن التنفيذ ms	عدد العقد	زمن التنفيذ ms	عدد العقد	الزمن
Noun_Adjective	2593	3270	204	2170	125	34%	39%
Adjective_Preposition	5612	5725	328	2318	218	60%	34%
Preposition_Verb	7339	7446	343	1739	203	77%	41%
Adjective_Adjective	7146	7694	356	2998	234	61%	34%
Verb_Adjective	14839	15481	546	3750	375	76%	31%
Preposition_Adjective	15564	15735	675	5208	420	67%	38%
Verb_Noun	17538	19423	718	6794	568	65%	21%
Verb_Preposition	26348	26544	1033	7760	675	71%	35%
Verb_Verb	29337	29076	908	6422	562	78%	38%
Preposition_Noun	50555	50819	2420	16870	1638	67%	32%
Noun_Preposition	68014	68272	4617	18750	3556	73%	23%
Noun_Verb	67460	69555	2817	15755	1740	77%	38%
Noun_Noun	77882	85963	3350	34344	2028	60%	39%
Adjective_Noun	97234	103361	5151	20419	3073	80%	40%

يظهر الشكلان (7) و(8) مقارنة بين خوارزمية FP-Growth والخوارزمية المقترحة من حيث متوسط زمن التنفيذ وعدد العقد مع النسب المئوية للتحسين:



الشكل(7): مقارنة بين خوارزمية FP-Growth والخوارزمية المقترحة من حيث زمن التنفيذ



الشكل(8): مقارنة بين خوارزمية FP-Growth والخوارزمية المقترحة من حيث عدد العقد

بالمقارنة نجد أن الخوارزمية المقترحة أسرع من خوارزمية FP Growth، لأنه يتم المرور مرة واحدة على قاعدة البيانات لبناء المخطط ودون الحاجة لحساب تكرار العقد وتخزينه، كذلك يتم الحصول على قواعد الترابط ممثلة بقواعد النموذج الشرطي دون بناء أشجار FP الشرطية (كما في خوارزمية FP-growth)، ومن الملاحظ أن عدد العقد في الخوارزمية المقترحة أقل من خوارزمية FP Growth لأن المخطط يحوي عدد عقد مساوي للعناصر المتكررة لأول مجموعة (itemsets -1) اي لا يوجد أي تكرار في عقد المخطط، بالتالي الذاكرة اللازمة لتخزين المخطط اقل. وفيما يلي جدول مقارنة للخوارزمية المقترحة بالاعتماد على المخطط مع عدة خوارزميات تنقيب أخرى، عند تنفيذها على مثالنا في الجدول (5):

الجدول (9): مقارنة بين الخوارزمية المقترحة وخوارزميات تنقيب اخرى

الخوارزمية / الصفات	Apriori	Vertical data forma	FP Growth	الخوارزمية المقترحة
مسح قاعدة	عدة مرات	عدت مرات	مرتين فقط	مرة واحدة فقط
التقنية	Breadth first search	Depth first Search	فرق تسد	فرق تسد
بنية التخزين	Array	Array	Tree	Graph
الزمن	38 ms	29 ms	22 ms	16 ms
العيوب	مجموعة العناصر المرشحة كثيرة جداً والمرور المتكرر على قاعدة البيانات يتطلب زمن ومساحة ذاكرة كبيرة	تتطلب ذاكرة إضافية لتخزين القوائم الوسيطية التي تضم ارقام أسطر العناصر	شجرة FP tree مكلفة في البناء وتحتاج الى مذيد من ذاكرة	محدودية استخدامها في المجالات التي تهتم بترتيب الكلمات كمجال تحليل المعنى

8. الاستنتاجات والتوصيات:

تم في هذا البحث اقتراح تعديل خوارزمية FP Growth لإيجاد قواعد المعنى، نظراً لأن هذه الخوارزمية تعطي قواعد ترابط تتجاهل تسلسل ورود الكلمات ضمنها، فتم الحصول على الخوارزمية المقترحة بالاعتماد على المخطط التي تعتبر خوارزمية فعالة في مجال تحليل المعنى، لأنها تعطي قواعد ترابط تراعي تسلسل ورود الكلمات ضمنها والذي يعتبر امراً مهماً في تحليل المعاني، كما انها تقترح التخلص من المسح المتكرر لقاعدة البيانات واستخدام المخطط الموجه بدلاً من الشجرة، بالتالي تخفيض مساحة الذاكرة المطلوبة لتخزين العناصر واختصار زمن البحث، من خلال تقليل عدد العقد، ومن ثم استخدام قواعد المعنى الناتجة عن الخوارزمية في التحقق من صحة معاني جمل اللغة الإنكليزية، بالتالي تم في هذا البحث تقديم نموذج لتوليد قواعد المعنى، يعتبر نظاماً قابلاً للتعلم لأنه من الممكن تحديث وإعادة بناء هذا النموذج عند توفر بيانات جديدة مضافة إلى قاعدة البيانات، وهذا النموذج يجعل من الحاسب أداة تحاكي الانسان الخبير في مجال اللغة الإنكليزية ويفتح آفاق مستقبلية لاستثماره في مجال اللغة العربية.

9. المراجع:

[1] Han, J., Pei,J., &Tong, H. (2022) .*Data Mining: Concepts and Techniques*. 4nd ed, Morgan Kaufmann, Elsevier Inc, United States of America,752 pages.

[2] Witten, L., Frank, E., & Hall, M. (2011). *Data Mining Practical Machine Learning Tools and Techniques*, 3th ed, Elsevier Inc, United States of America,665 pages.

[3] Omar, N., Hanna, P., & Mc Kevitt, P. (2006, June). *Semantic Analysis in the Automation of ER Modelling through Natural Language Processing. In 2006 International Conference on Computing & Informatics* . IEEE

[4] Osipov, G., Smirnov, I., Tikhomirov, I ., & Vybornova, O . (2012, September). Technologies for Semantic Analysis of Scientific Publications. In 2012 6th IEEE International Conference Intelligent Systems. IEEE

[5] Lakshmi, K. S., & Kumar, G. S. (2014). *Association rule extraction from medical transcripts of diabetic patients. The Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014).*

[6] دقاق، مصطفى، الخضري، & أمل.(2016). *تحسين نتائج استرجاع المعلومات العربية دلالياً باستخدام الأنتولوجيا. مجلة جامعة البعث للعلوم الهندسية*،38(46).

[7] Bhujade, V., & Jonwe, N.J. (2011, October) . *Knowledge Discovery in Text Mining Technique Using Association Rules Extraction. in 2011 International Conference on Computational Intelligence and Communication Networks* (pp. 498-502). IEEE.

[8] Yamuna Devi, N., & Devi Shree,J. (2013). *A novel approach and comparative study of association rule algorithms in validation of semantics of sentences, International Journal of Computer Applications* ,France, Vol 62 , No 3 , 22-26.

[9] Vij, S., Jain, A., Tayal, D., & Castillo, O. (2017). *Fuzzy Logic for Inculcating Significance of Semantic Relations in Word Sense Disambiguation Using a WordNet Graph. International Journal of Fuzzy Systems*, 20(2) , 444 – 459.

[10] Luo, X., & Chen, Z .(2020). *English text quality analysis based on recurrent neural network and semantic segmentation*. Future Generation Computer Systems, vol.112 , 507–511.

[11] Nadkarni, P. M., Ohno-Machado, L., & Chapman, W. W. (2011). *Natural language processing: an introduction. Journal of the American Medical Informatics Association, 18(5), 544–551.*

[12] Daves, M. (2011). N-grams data from the Corpus of Contemporary American English (COCA), Downloaded from http://www.ngrams.info on May 04, 2020.

[13] Du, J., Zhang, X., Zhang, H., & Chen, L. (2016, May). *Research and improvement of Apriori algorithm. 2016 Sixth International Conference on Information Science and Technology (ICIST).*

[14] Guo, Y., & Wang, Z. (2010, March). *A vertical format algorithm for mining frequent item sets. 2010 2nd International Conference on Advanced Computer Control* (pp.11-13). IEEE.

[15] Singh, A. K., Kumar, A., & Maurya, A. K. (2014, May). *An empirical analysis and comparison of apriori and FP- growth algorithm for frequent pattern mining*. 2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies (pp.1599-1602). IEEE.

[16] ديوب، يعرب.(2020). *تسريع التحليل النحوي باستخدام النحو الشكلي الاحتمالي المعدل. مجلة جامعة طرطوس للبحوث والدارسات العلمية–سلسلة العلوم الهندسية*، 4(9).

[17] ديوب، يعرب.(2020). *التحويل الكلي للنحو الشكلي المبرمج المقيد الى النحو المبرمج الحر. مجلة جامعة طرطوس للبحوث والدارسات العلمية–سلسلة العلوم الهندسية*، 4(6).

[18] Jamsheela, O., & Raju, G. (2015, June). *Frequent itemset mining algorithms: A literature survey. 2015 IEEE International Advance Computing Conference (IACC).*

[19] Welborn, C., & Rudolph, G. (2020, October). *Formal Definitions for Common Data Structures and Algorithms. in 2020 Intermountain Engineering*, Technology and Computing (IETC).

[20] Borah, A., & Nath, B. (2019). *Tree based frequent and rare pattern mining techniques:a comprehensive structural and empirical analysis. Springer Nature Applied Sciences, Switzerland AG.*